INTEGRATING RULE-BASED AND NEURAL-NET TECHNIQUES FOR SPECTRAL ANALYSIS

THESIS

Arthur L. Sumner
First Lieutenant, USAF

AFIT/GCE/ENG/90J-01

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GCE/ENG/90J-01

INTEGRATING RULE-BASED AND NEURAL-NET TECHNIQUES FOR SPECTRAL ANALYSIS

THESIS

Arthur L. Sumner
First Lieutenant, USAF

AFIT/GCE/ENG/90J-01

Approved for public release; distribution unlimited

AFIT/GCE/ENG/90J-01

INTEGRATING RULE-BASED AND NEURAL-NET TECHNIQUES FOR SPECTRAL ANALYSIS

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Computer Engineering

Arthur L. Sumner, B.S.C.S.

First Lieutenant, USAF

June 1990

Preface

The purpose of this study was to develop an integrated rule-based expert system and neural network to automatically determine both the atomic and the molecular species in raw spectra. This research was performed at the Air Force Institute of Technology (AFIT) with assistance from other organizations at Wright-Patterson Air Force Base, Ohio.

In developing the rule-base and neural network, and in writing this report, I had a great deal of help from others. I am deeply indebted to my faculty advisor, Steven K. Rogers, for all his assistance and direction. I am also greatly indebted to L. Grant Miller for all his insight into the spectral analysis process and to Peggy J. Grigsby for her aid in making the preliminary identification of atomic and molecular species. I wish to thank Greg Tarr who provided the neural network simulator on which this work was carried out and Frank M. Brown who provided assistance during the development of the rule-based expert system. I also wish to thank those who reviewed this work: Bonnie L. Keeler, C. Todd Hawley, and Stephen J. Powers. Finally, I wish to thank my wife, Suzanne, for her understanding and her devotion to me during the course of this thesis research.

Arthur L. Sumner

Table of Contents

List of Figures

List of Tables

AFIT/GCE/ENG/90J-01

Abstract

Spectral analysis involving the determination of atomic and molecular species present in multi-spectral data is a very time consuming task, especially considering the fact that there are typically thousands of spectra collected during each experiment. Due to the overwhelming amount of available spectral data and the time required to analyze these data, a robust automatic method for performing preliminary spectral analysis is needed. This research focused on the development of a rule-based expert system and a supervised artificial neural network with error correction learning, specifically a three-layer, feed-forward, back-propagation perceptron. The objective was to develop an integrated spectral analysis system which would perform preliminary spectral analysis and save the analysts from the task of reviewing thousands of spectral frames. The input to the neural network, which is screened by the rule-base, is raw spectral data, with the output consisting of the classification of both atomic and molecular species in the source.

The idea that computers can perform the spectral identification of molecular and atomic radiators goes back to the earliest days of computers. Most commercially available spectral identification programs use some type of statistical pattern recognition technique containing a database of known species. One of the greatest advantages of neural networks over traditional pattern recognition techniques is the ability

of the neural networks to learn. Once a neural network has been trained, it may be able to make classifications with a general amount of robustness not normally found in statistical recognizers. Another major advantage of neural networks over statistical methods is that new species can be recognized by simply presenting samples of spectra to the neural network, eliminating the time-consuming task of trying to statistically characterize the new species. Neural networks are not able to recognize every species and will never replace the analyst, but they can reduce the time required for preliminary analysis and aid the analyst in picking out species that a statistical technique might have missed.

INTEGRATING RULE-BASED AND NEURAL-NET TECHNIQUES FOR SPECTRAL ANALYSIS

I. Introduction

General Issue

Spectral analysis involving the determination of the atomic and molecular species present in a frame of multi-spectral data is a very time consuming and monotonous task. It involves a long, hard, and concentrated effort by spectroscopists who must be very familiar with the radiating characteristics of elements and molecules. The time it takes to train spectral analysts makes them a valuable and scarce commodity. Many organizations are not able to employ enough analysts to analyze the data that have been collected. For this reason, a great deal of the data which have been so carefully obtained sits on the shelf waiting to be analyzed.

The problem is compounded by the fact that there are a great deal of data collected during each experiment. The high sample rates which are used in experiments, several times a second, coupled with the duration of the experiment, sometimes many hours, results in thousands of spectral frames which cannot be analyzed in their entirety by trained human analysts.

This problem is usually solved by either averaging hundreds of the frames together or by scanning through the data for frames that have a large number of lines and bands. This presents a problem because events

that occur only over a very short time might be missed in the averaging process or skipped altogether if a specific frame is not chosen. To obtain the full value of the data, a method to efficiently analyze all the data must be found.

Justification

Several organizations at Wright-Patterson Air Force Base are involved in the analysis of raw multi-spectral data and have an enormous amount of these data that must be analyzed in a timely manner. Analysis of much of these data is not performed due to a lack of available manpower.

Many experiments are conducted each year. Each of these experiments collects about five thousand frames of spectral data, each taken at a different point in time. Each frame contains ten rows, or ten spatially distinct regions of the field of view which need to be analyzed. There is little wonder that much of the data from an experiment is skipped. Many of the experiments are not even examined due to the overwhelming workload.

Currently, only about two-thirds of the experiments are analyzed and of these experiments, only one-tenth of the spectral frames are analyzed. It currently takes approximately 50 hours for one set of spectral data to be scanned to find appropriate regions for further analysis, and another 50 hours for the detailed analysis of these selected regions. This requires that vast amounts of data which might reveal important and interesting results about an experiment be ignored. This 100 hours is usually expended over more than a month because of the

monotonous nature of such spectral analysis. This leaves a large gap in the data analysis which requires correction. For this reason, the organizations which are involved in this type of spectral analysis desire some way to analyze the data more efficiently.

Problem Statement

Due to the overwhelming amount of available spectral data and the time required to analyze these data, an automatic method for doing at least some preliminary spectral analysis is needed so that the workload of the analysts and the backlog of spectral data will be reduced.

Research Objectives

This research focused on the development of a rule-based expert system and a supervised artificial neural network with error correction learning. The objective was to develop an integrated rule-base and neural network which would perform the preliminary spectral analysis and save the analysts from the mundane task of reviewing thousands of spectral frames. The input to the neural network, which was screened by the rule-base, was unclassified raw spectral data, with the output consisting of the classification of both atomic and molecular species in the source.

Research Questions

The foremost question of this research was to determine if a neural network could be developed which would efficiently and accurately perform the preliminary spectral analysis for the organizations which are involved in analyzing this type of multi-spectral data. The

secondary question was to determine if an expert system could be integrated with this neural network to aid in the recognition process by filtering the spectral data before it was presented to the neural network.

Definitions

A frame of spectral data refers to ten snapshots of spectra, each taken simultaneously at a discrete point in time as shown in Figure 1. A row of spectral data refers to one of the ten different snapshots of spectra taken at one discrete time which make up a frame. These ten rows are stacked on top of each other in the sensor providing some indication as to the vertical location of the radiating species within the field of view.

An optical diagram of the sensor showing the relative position of the ten rows is presented in Figure 2. Each row represents the spectral dispersion of a point source into the horizontal direction. As shown by the diagram of the charge coupled device (CCD) in Figure 3, these ten rows are stacked on top of each other, row 1 at the top and row 10 at the bottom. The metric sensor, depicted in Figure 2, collects integrated broadband spectra of all ten rows combined.

Three different types of rows have been defined: good, bad, and background. A good row, for example rows 1 and 3 of Figure 1, is one which contains spectra from the object being viewed and does not have a range of saturated pixels. A bad row is one which has a range of saturated pixels as in row 2 of Figure 1. A background row is one in which only the background is present, as in rows 4, 5, 6, 7, 8, 9, and

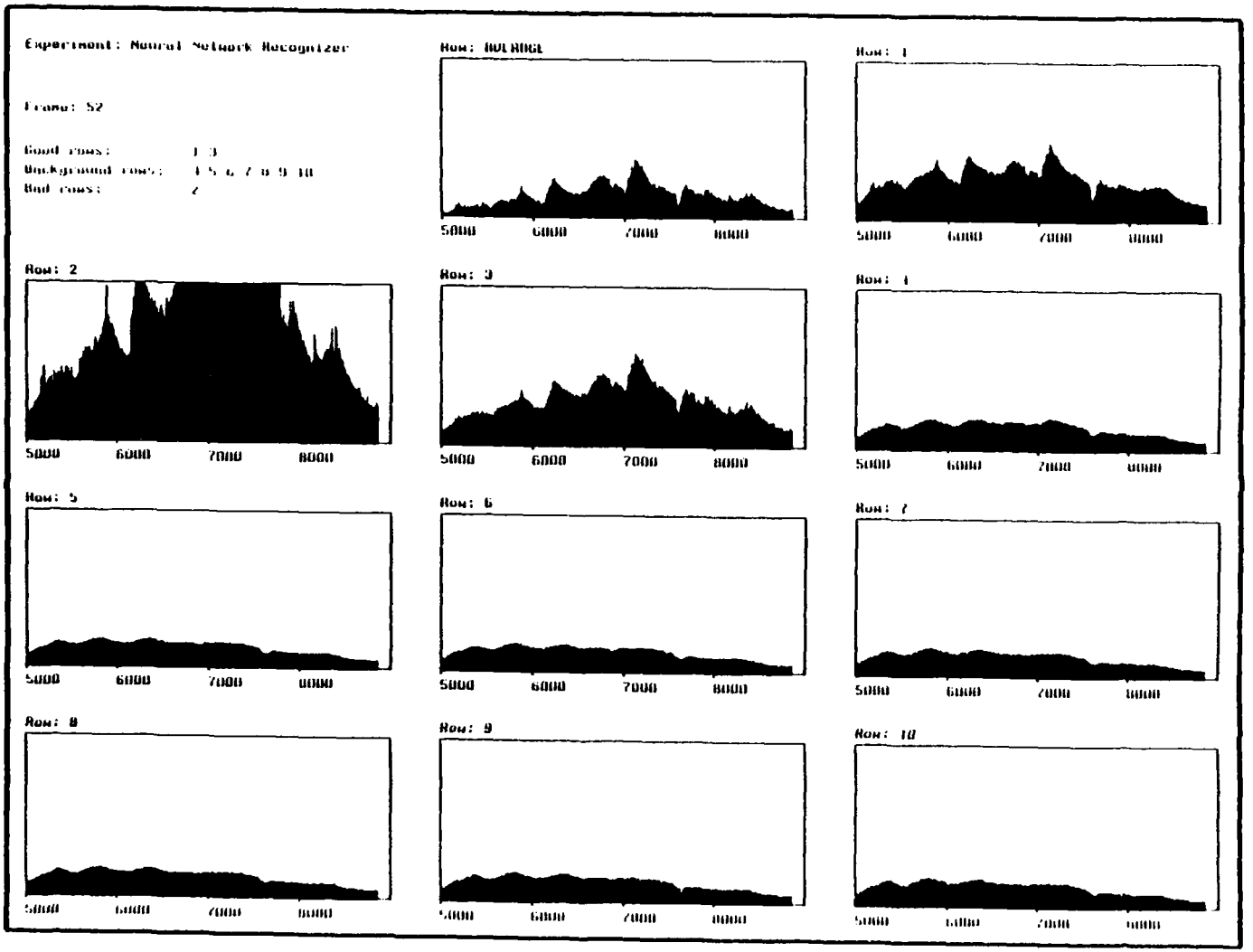

Figure 1. Sample Spectral Frame

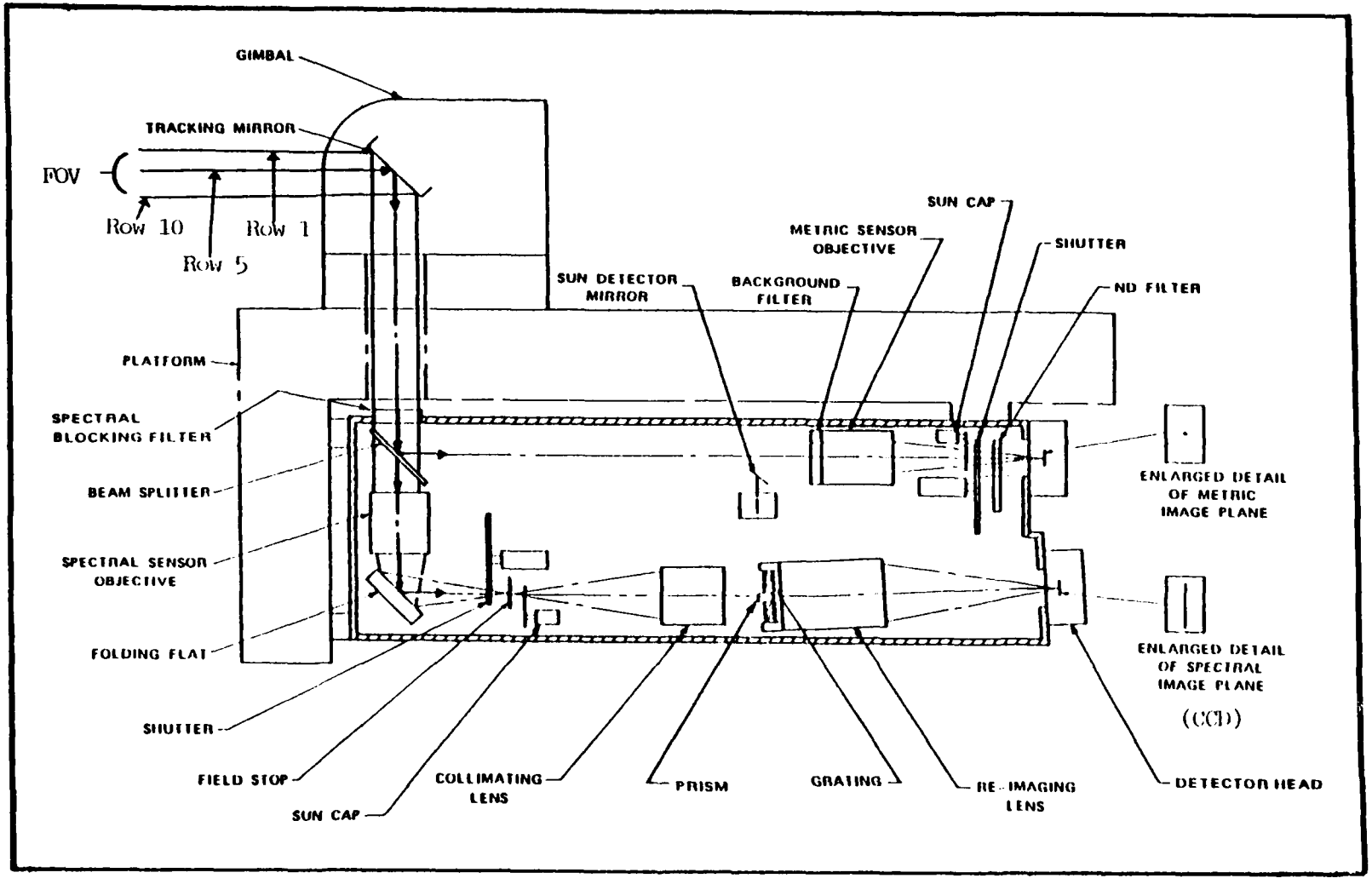

Figure 2. Sensor Optical Diagram (12:82)

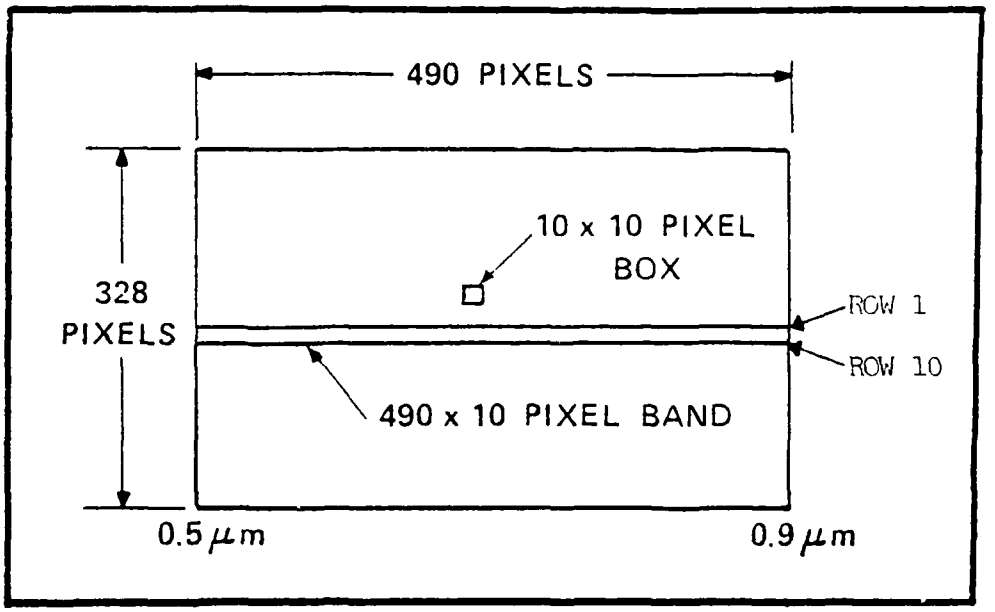

Figure 3. Charge Coupled Device (CCD) Layout (12:86)

10 of Figure 1. The average row is created by subtracting the average of the background rows from the average of the good rows. The good rows are used by the neural network for training and will be used for species recognition once the neural network is implemented in the spectral analysis software.

There are two broad classes of species which have been successfully recognized by the neural network: atomic and molecular. Atomic species refer to the species which radiate due to the excitation of atoms, such as sodium (Na) and potassium (K), while molecular radiators refer to the species which radiate due to the excitation of molecules, such as aluminum oxide (AlO) and titanium oxide (TiO). Atomic species are characterized by lines (a small number of pixels excited), while

molecular species are characterized by bands (a much larger number of pixels excited). Figure 4 presents examples of both atomic and molecular species.

Assumptions

A number of assumptions were made about the data used in this project. The data were not corrected for atmospheric transmission and there may be some spectral artifacts due to effects of the atmosphere. According to Dr L. Grant Miller, a trained spectroscopist working for the Air Force, the program that makes atmospheric corrections, LOWTRAN 7, has a tendency to over-correct and introduce false lines and bands into the spectra (11). For this reason, uncorrected spectra were used for neural network recognition. If LOWTRAN 7 is added in the future, the neural network may be able to provide a greater recognition rate. The neural network should eventually be able to learn around the over-correction since the over-correction would be constant and since one of the great advantages of artificial neural networks is the ability to pick out features from raw data (13).

The range from the sensor to the experiment varied from experiment to experiment. The difference is less than an order of magnitude and therefore was assumed to be negligible. According to Dr Miller, LOWTRAN 7 would have taken into account variations due to sensor range; however, the tendency to over-correct and introduce false lines and bands into the spectra makes the correction of questionable value (11). If LOWTRAN 7 is used in the future, the range difference will be taken into account at the same time that the atmospheric transmission

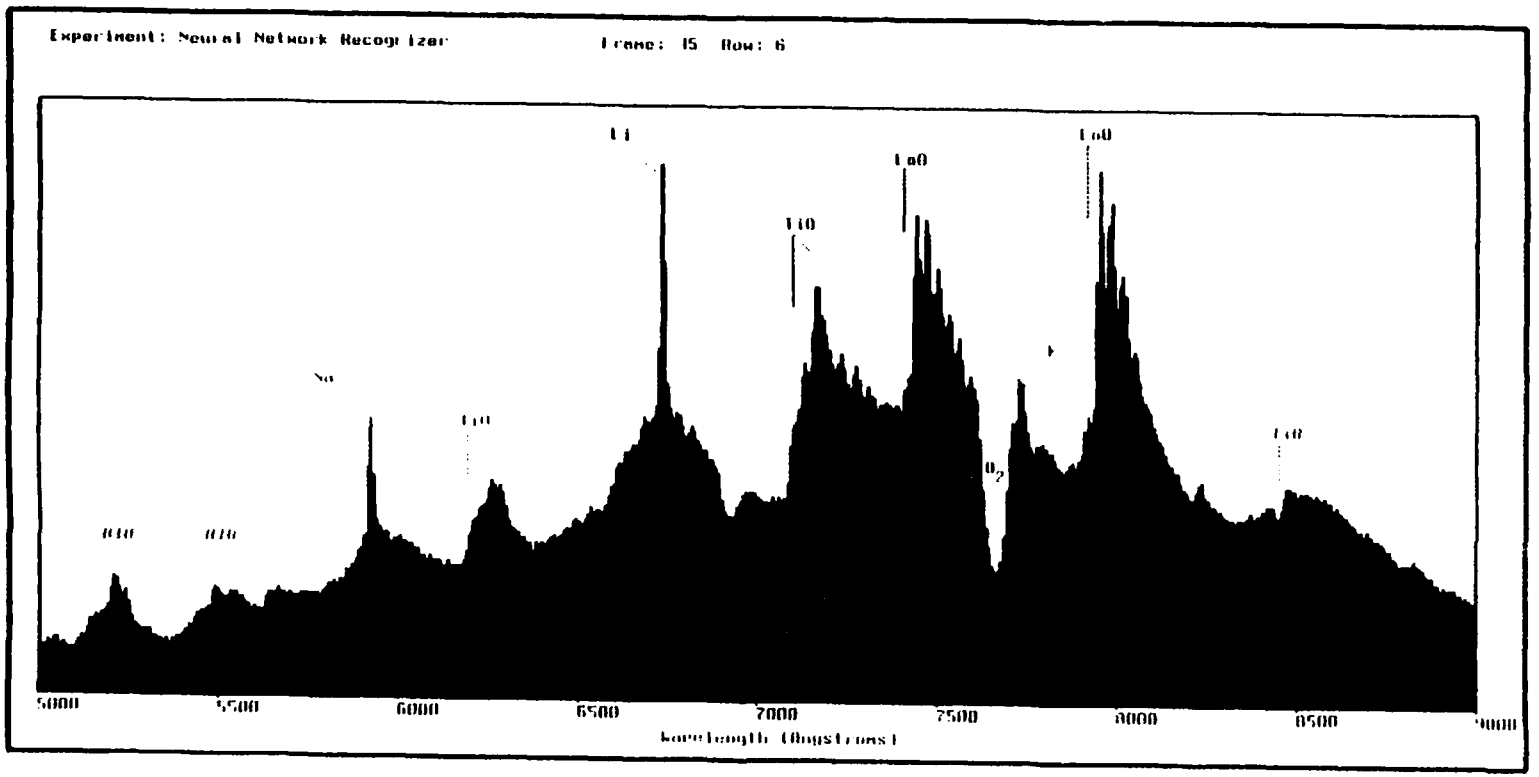

Figure 4. Labeled Spectral Row

correction is performed. Corrections for sensor response, which require only the subtraction of the sensor dark current levels, were also dismissed because the variations between sensors are believed to be negligible and the neural network performed at an acceptable level without the corrections.

Background radiation was not subtracted out but this capability may be added in the future by averaging spectral rows with no data and subtracting this average from the rows which contain spectral data. Since experiments were performed at different times of the day, this should remove differences caused by the sun during daytime experiments. Rows of data where the sensor was saturated were ignored and labeled as bad rows. According to Dr Miller, although saturated rows may contain regions with valid species, ignoring these rows is the preferred method for dealing with this problem (11).

Scope

This project entailed the development of an integrated rule-based expert system and neural network for the preliminary analysis of spectral data. This spectral analysis system and supporting interface were developed on a Silicon Graphics workstation which was available at the Air Force Institute of Technology (AFIT). It was developed in such a way as to facilitate the easy integration of the integrated rule-base and neural network into the enhanced spectral analysis software which is currently under development on the new Silicon Graphics IRIS 4-D workstations.

The CCD, pictured in Figure 3, contains ten rows of pixels designated to acquire spectral data. These ten rows of pixels correspond to the ten spatially distinct rows seen in each frame. Each row is made up of 490 pixels, with each pixel spaced .0008 microns apart. The bandpass of the sensor is from .5 to .9 microns (the visible to short-infrared range of the electromagnetic spectrum), with a resolution of .0016 microns. The 10 x 10 pixel box on the CCD is used for coarse imaging of the object to aid in the initial wavelength assignment of the spectra.

Initially, the neural network was trained to identify the alkali metals most commonly seen in the spectral data: sodium (Na), lithium (Li), and potassium (K). Since the wavelength range of molecular bands is broader than for atomic lines, the molecular species identification problem is more difficult. The neural network was trained on three molecular species - aluminum oxide (AlO), lanthanum oxide (LaO) and titanium oxide (TiO) - and will eventually be trained on a larger group of molecular species including calcium oxide (CaO and Ca_2O_2), calcium chloride (CaCl), calcium fluoride (CaF), calcium hydroxide (CaOH), chromium oxide (CrO), iron oxide (FeO), and other species which may be found in future spectra. Figure 4 presents a sample row which contains the six species which were recognized as part of this project.

Limitations

The objective of this project was not to develop a spectral analysis system which would point out exactly where spectral features lie in a frame of spectral data. Rather, the focus was on developing an

integrated rule-base and neural network which would determine which frames in a sequence of thousands contain certain atomic and molecular species. These data can then be displayed in a time versus species history graph. This gives the analysts a good place to start when reviewing spectra for detailed examination.

Summary

This project is one which will be of great value to spectral analysts in the Air Force as well as to others who are concerned with spectroscopy. The great quantity of spectral data available makes it imperative that some new method of preliminary analysis be developed. The use of a rule-base and neural network to perform this analysis will greatly increase the efficiency of the spectral analysis process.

The following chapter includes background information relating to work dealing with neural networks, expert systems, and spectral analysis. The methodology used in the research and development of the integrated rule-base and neural network and the results achieved are reviewed in the next two chapters, with the final conclusions and recommendations in the last chapter.

II. Background

In the past few years, there have been a great many strides made in the area of machine-aided spectral analysis. The idea that computers can perform the spectral identification of atomic and molecular radiators goes back to the earliest days of computers. There has always been a push to find easier ways to perform spectral analysis, and this drive towards automation of such analysis will probably not end until a computer does it all.

A report produced by the Lowell Technological Institute Research Foundation back in 1962 detailed a computer program which would determine elements present in a sample. This program took known spectral lines and compared them with unknown spectra. Through a system of normalization and distance measurements, the unknown species were then compared to all the spectral lines in the spectral catalogue. The smallest distance between the unknown species and a known species line was displayed as the species identification. (17:6-8)

This system of making normalized distance measurements between an unknown spectral line and a catalogue of known spectral lines is still used today as the primary means of automated spectral recognition. Most commercially available spectral identification programs use some type of statistical pattern recognition technique utilizing a database of known species. Both Sprouse Scientific Systems, Inc. and Sadtler Research Laboratories market a spectral recognition program which uses some form of normalized distance measurements to identify species. Both programs offer the standard feature of displaying probabilities based on the

distance computed as well as other options such as subtraction of already identified species. Sprouse Scientific Systems, Inc. also offers a feature which displays the molecular structure of molecular species. These systems are widely used in research and are very useful in performing spectral analysis. (10:1-5; 18:2-3)

Despite the usefulness of such automated systems in certain situations, there are problems with all types of pattern recognition systems which identify species. One of the largest problems is that of noise. The standard procedure for identifying spectral features requires a set of calibrated or reference spectra which is acquired under laboratory conditions. Such a set of spectra obviously will contain less noise than the spectrum which contains the unknown species. This can cause major problems when it is time to identify unknown species in raw spectra.

There are two distinct classes of noise. One class is standard additive Gaussian noise and the other is non-Gaussian noise (7). The fact that the spectra are taken at varying distances is a good example of a source of Gaussian noise according to Dr. Miller, since the signal changes proportionally to the distance involved (11). The energy normalization process used in statistical recognizers is the means to reduce the effect of additive Gaussian noise, since the normalized data should have the noise scaled to the same level (7). Non-Gaussian noise is another problem entirely, and may show up as artificial spectral lines or as a partially or completely saturated frame of spectra. This makes it very difficult to handle automatically, and therefore it

usually requires discrimination by a trained spectral analyst. Differing sensor sensitivities are one source of non-Gaussian noise.

There are many methods of energy normalization, but one of the most common is to divide each spectral pixel by the square root of the sum of each pixel squared. The formal definition of the equation is as follows:

$$N(\mu_\iota) = \frac{\mu_\iota}{\left[\sum_{\kappa} \mu_\iota^{\,2}\right]^{\frac{1}{2}}} \qquad (1)$$

where

$N(\mu_\iota)$ = normalized pixel intensity at wavelength ι
μ_ι = pixel intensity at wavelength ι
κ = pixel number in the row

Although it may take more processing time than some energy normalization equations, this equation generally provides the best results. (8:7)

Once the data have been normalized, a distance measurement must be made to determine which reference spectrum most closely matches the unknown spectral feature. A feature space, or a set of input parameters, must be selected from which to take these distance measurements (7). This may be as simple as the raw spectra, or it may include a complex transform of the spectra, such as a Fourier transform or a Laplace transform. The actual feature space used by commercial software is usually proprietary and it is difficult to determine the best feature space to use. The intensities at each wavelength, or the components of the transform, are then made into a vector. Determination of a feature space can be automated if an artificial neural network is used as the classifier (15:1-12).

The distance between vectors is then calculated by using a distance metric. Again, as with the normalization procedure, there are a number of different methods for determining the distance between the vectors. One of the most common is the Minkowski (M) distance. There are several different Minkowski distances, but the general equation is as follows:

$$D_{12} = \left[\sum_{\kappa} \left| V_{1\kappa} - V_{2\kappa} \right|^{M} \right]^{1/M} \qquad (2)$$

where

D_{12} = distance between vectors V_1 and V_2
κ = vector component
V_1 = unknown spectra vector
V_2 = reference spectra vector
M = Minkowski number

The Minkowski number need not be a positive number or even an integer. There are many interesting effects that occur by changing the Minkowski number. The Minkowski distance most often used is where M=2 (Euclidean distance), but the Minkowski distance where M=1 (taxi distance) is more efficiently computed since it requires fewer mathematical operations. Minkowski numbers which give better distributions of the items being recognized over the feature space may be selected through trial and error. (7)

Once the distances between the unknown species and all the reference spectra have been computed, they are placed in a distance matrix. The match is then simply a choice of the smallest distance. A threshold may also be applied so that matches greater than a predetermined amount may be discarded. This helps to prevent the

matching of a known species to an unknown species that has no corresponding spectra in the spectral catalogue. (7)

Although these methods work relatively well and seem to be the most commonly used today, statistical pattern recognition techniques for the automatic determination of species contained in unknown spectra do have limitations. These methods are prone to errors, as are all pattern recognition techniques including those based on neural networks. These errors cause results to vary depending upon the noise present and the feature space chosen. This potential for error requires that a more robust method for spectral identification be developed, one that will make the choice of the feature space as well as automatically take care of any normalization problems. In addition, it should possess the capability of identifying new species by simply presenting samples of spectra to the recognizer, therefore eliminating the time consuming task of statistically characterizing the new species.

What method can be used to make spectral identifications, automatically choose the feature space, and reduce the influence of additive Gaussian noise? The new, rapidly growing field of artificial neural networks may provide the answer. A neural network is a collection of neurons, a simplified model of those found in the human brain. An artificial neural network is a neural network which is made out of computer chips or written in software. Artificial neural networks can even be made of lenses and holograms. Called optical neural networks, these neural networks may help with the automatic spectral identification problem in the future by directly processing the spectra. (3:1573-1581; 14:5-26)

An artificial neuron is basically a device that accepts a number of inputs, weights each of the inputs, performs some nonlinear mathematical function on the sum of the weighted inputs, and outputs a single value. Figure 5 shows the schematic representation of a typical neuron. One theory in biological neural networks is that all brain functions are just calculations performed on a group of weighted inputs with the output produced by varying firing rates of the neurons. (13)

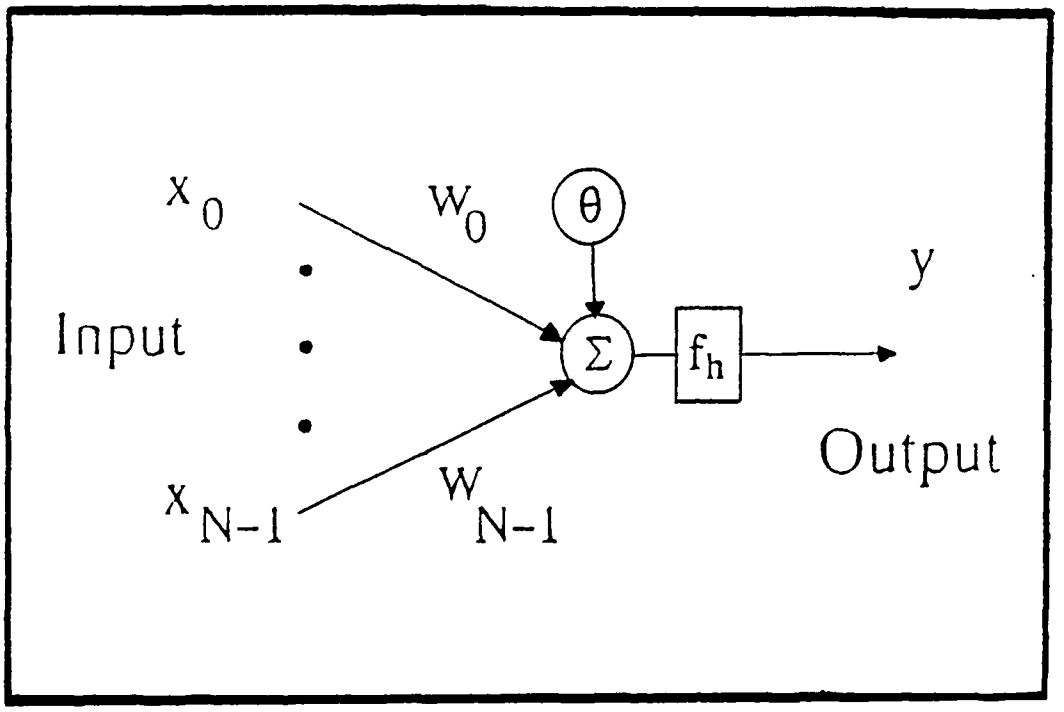

Figure 5. Artificial Neuron Schematic (14:28)

An artificial neural network is a group of neurons connected both serially and in parallel. An example of one of the more popular types of neural networks, the multilayer perceptron, is presented in Figure 6. This neural network is a generalization of the single layer perceptron

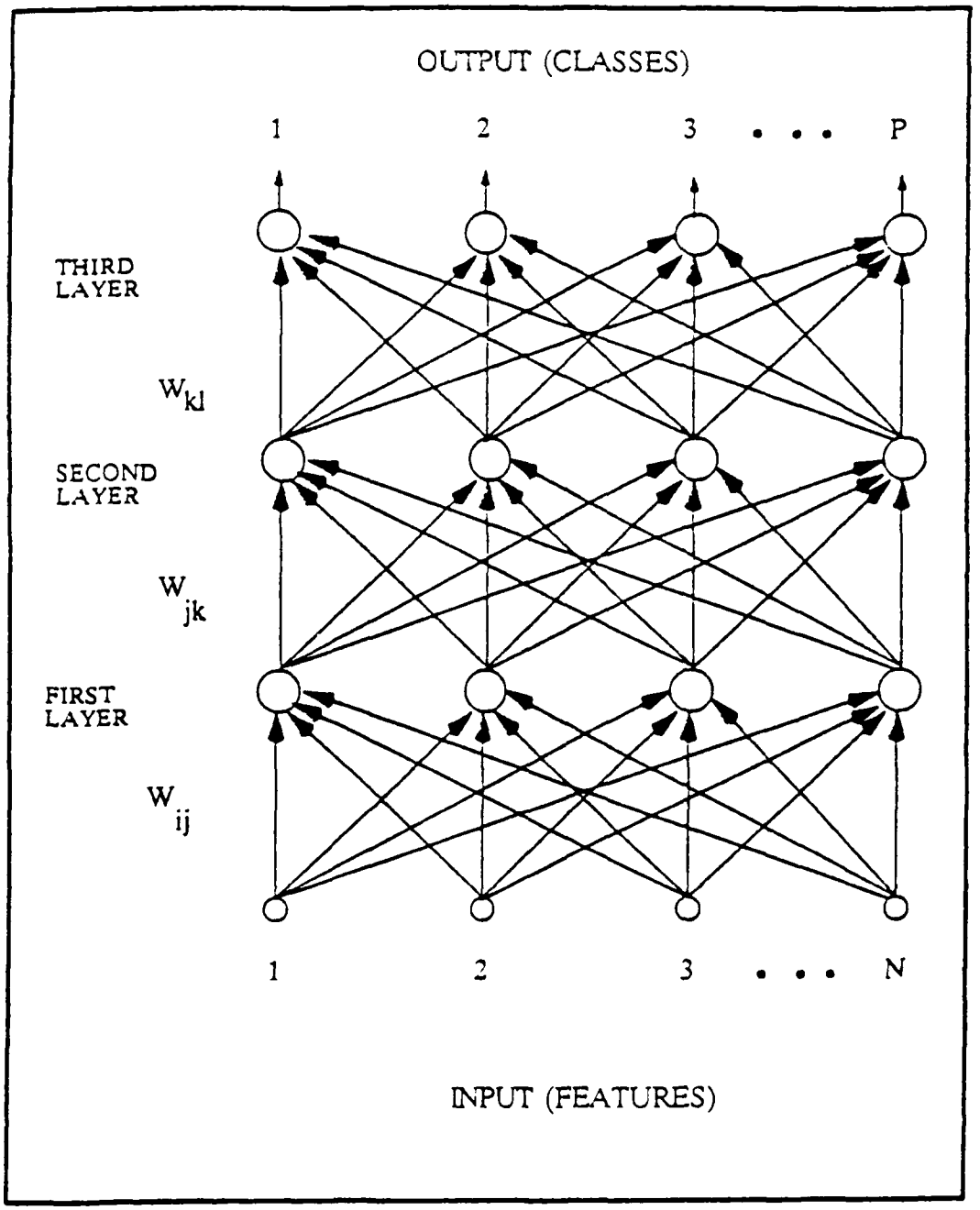

Figure 6. Multilayer Perceptron (14:38)

made possible by the advent of learning laws such as the back propagation algorithm (9:15-18). The inputs to a neural network are the data from the feature space of the item being identified. The identification is made by the neural network and the classification is determined by checking for the top layer neuron with the highest firing level. This winning neuron corresponds to the item being identified. (13)

One of the greatest advantages of a neural network over traditional pattern recognition techniques is the ability of the neural network to learn. Once a neural network has been trained, it may be able to make classifications with a general amount of robustness not normally found in statistical recognizers. Even though a neural network is mathematically a statistical recognizer, the training process may be able to statistically characterize the problem more accurately than do traditional techniques (16:1-9). The difference between traditional pattern recognition techniques and neural networks lies in the determination of the feature space. A neural network is able to determine the important features from the raw data which it is fed. It is able to determine the feature space because it is a type of associative memory. (13)

The great advantage of neural networks is their ability to be trained. This can be accomplished by a back propagation training algorithm. In a multilayer perceptron, the training involves several steps. The weights are first randomly set. An input vector is fed into the input nodes and the resulting output is calculated for the first layer using the equation:

$$y_\kappa = f_h\left[\sum_\iota \left[w_{\iota\kappa} x_\iota\right] + \theta_\kappa\right] \quad (3)$$

where

y_κ = output value for node κ
f_h = nonlinear transfer function
ι = node from layer below current layer
$w_{\iota\kappa}$ = weighting factor from node ι
x_ι = input value from node ι
θ_κ = offset for node κ

The output from the first layer is then used as the input to the next layer. Equation 3 is again used to calculate the outputs for the second layer. This process continues recursively until the top layer's outputs have been calculated. (9:16)

The back propagation algorithm is then invoked to update the weights beginning with the top layer. The desired value at the top layer is compared to the resulting value, and a correction to the weighting factor is calculated using the equation:

$$w_{\iota\kappa}(t + 1) = w_{\iota\kappa}(t) + \eta\delta_\kappa x_\iota \quad (4)$$

where

$w_{\iota\kappa}(t + 1)$ = new weighting factor from node ι
$w_{\iota\kappa}(t)$ = old weighing factor from node ι
η = gain factor
δ_κ = error term for node κ
x_ι = input value from node ι

When the node is an output node, the equation for δ_κ is as follows:

$$\delta_\kappa = y_\kappa(1 - y_\kappa)(d_\kappa - y_\kappa) \quad (5)$$

where

δ_κ = error term for node κ
y_κ = actual output value of node κ
d_κ = desired output value of node κ

When the node is an internal node, the equation for δ_κ changes. An internal node is different in that it is difficult to determine what the desired value for the internal nodes should be. The solution to this problem is to calculate δ_κ using the following equation:

$$\delta_\kappa = y_\kappa(1 - y_\kappa) \sum_\iota \left[\delta_\iota w_{\kappa\iota}\right] \tag{6}$$

where

δ_κ = error term for some hidden node κ
y_κ = actual output value for hidden node κ
ι = nodes in layer above node κ
δ_ι = error term for node ι in layer above node κ
$w_{\kappa\iota}$ = weight to layer above node κ

The updating of the weights continues recursively until the first layer of the neural network is reached. Once the learning phase is over, the use of the back propagation algorithm is discontinued by the neural network. (9:17)

Once a neural network has been trained, it is able to make classifications with a general amount of robustness. It will not be able to pick up everything, but it can generalize to changes in the data which it has not seen before.

There have been a number of 'artificial intelligence' systems developed to perform spectral analysis. One system was developed which uses a digital learning network, similar in concept to neural networks (20:246-247). Another system uses a rule-based expert system to aid in spectral analysis (5:453-463). Neither system was designed to take raw spectra and determine which species are present, rather they were designed to take a wide variety of data which are available in a laboratory setting and classify the compounds being analyzed.

There has been little research on the specific use of neural networks for pattern recognition of spectra. One project utilized an optical neural network to determine minerals in data gathered by an imaging spectrometer. This was successful in picking out the different combinations of minerals present due to changes in the structure of the spectrum. However, since different minerals may change the structure of the entire spectrum rather than just distinct regions as with atomic and molecular radiators, adapting this technique to this thesis would ignore the a priori knowledge available and result in greater training and recognition times. (1:3129-3133; 2:541-546)

Another project which was reported at the International Conference on Neural Networks in 1988 used a neural network to classify chemicals seen in a spectrum. This project had only ten input nodes and four output nodes and used a single layer perceptron. The output of the neural network merely separated chemicals into four different groups for later analysis. (4:657)

Although this neural network which classifies chemicals into general groups helps in the spectral analysis process, it does not point out specific spectral features, such as atomic and molecular species. This is a much more ambitious project which requires the integration of a rule-based expert system with either a single neural network with a large number of input nodes, 490 for this project, and a large number of output nodes, or a number of individual neural networks, one for each species to be recognized. This is the subject of the rest of this thesis.

III. Methodology

Introduction

The development of an integrated rule-based expert system and back-propagation neural network for recognition of atomic and molecular species required several iterations in order to achieve an acceptable solution. Several neural network topologies were considered before deciding on a general solution. The difficulties encountered with these data were the result of the large number of pixels in each row of data, the huge variations in the data, especially the wavelength scale, and the different types of rows: good, bad, and background.

Row Management

The first problem was to eliminate any rows that were saturated or background. This was performed by developing a routine to integrate each of the rows of data. The row with the least total area was declared to be a background row. The remaining rows were then compared to the background row, pixel for pixel. If there was no more than a ±5 pixel count difference over any of the pixels in a row, the row was designated as a background row. Saturated rows were then determined by checking each non-background row to see if there was a region with six or more consecutive pixels that were greater than 250 pixel counts. These rows, marked as background or saturated, were ignored by the neural network both during training and during use as a recognizer.

Although this row management system was approximately 90% accurate, higher accuracy was desired. This required the development of an expert

system to handle the row management task. The original algorithmic approach to choosing the good, bad, and background rows suffered from many problems. There were only a couple of conditions that were integrated into this algorithmic approach due to the complex conditional clauses.

The possibility of using an expert system to make the determination of the background and saturated rows was explored because a large set of rules could be developed to handle the wide range of row conditions without regard to the structure of the conditional statements. The CLIPS ('C' Language Integrated Production System) expert system shell was chosen because it could be integrated with very little effort into the 'C' code in which the neural network and the spectral analysis package were developed (6:Sec II,1-32). The CLIPS row management rule-base is presented in Appendix A.

There are four basic facts that are fed to the CLIPS rule-base for use in determining the row dispositions. The AVERAGE-PIXEL facts give the average pixel intensity for each of the rows in a frame. The AREA-DEVIATION facts give the maximum deviation of a pixel from the corresponding pixel in the base row. The base row is the row with the least area after it has been smoothed using a seven-point averaging technique. The BAD-DATA facts flag each row which has a group of six or more consecutive pixels which are greater than 250 pixel counts. The BAD-DATA fact also flags each row which has a sudden drop in pixel intensity due to the intensity modulus characteristic of the CCD when saturated. The fact-list for the example frame presented in Figure 7 follows.

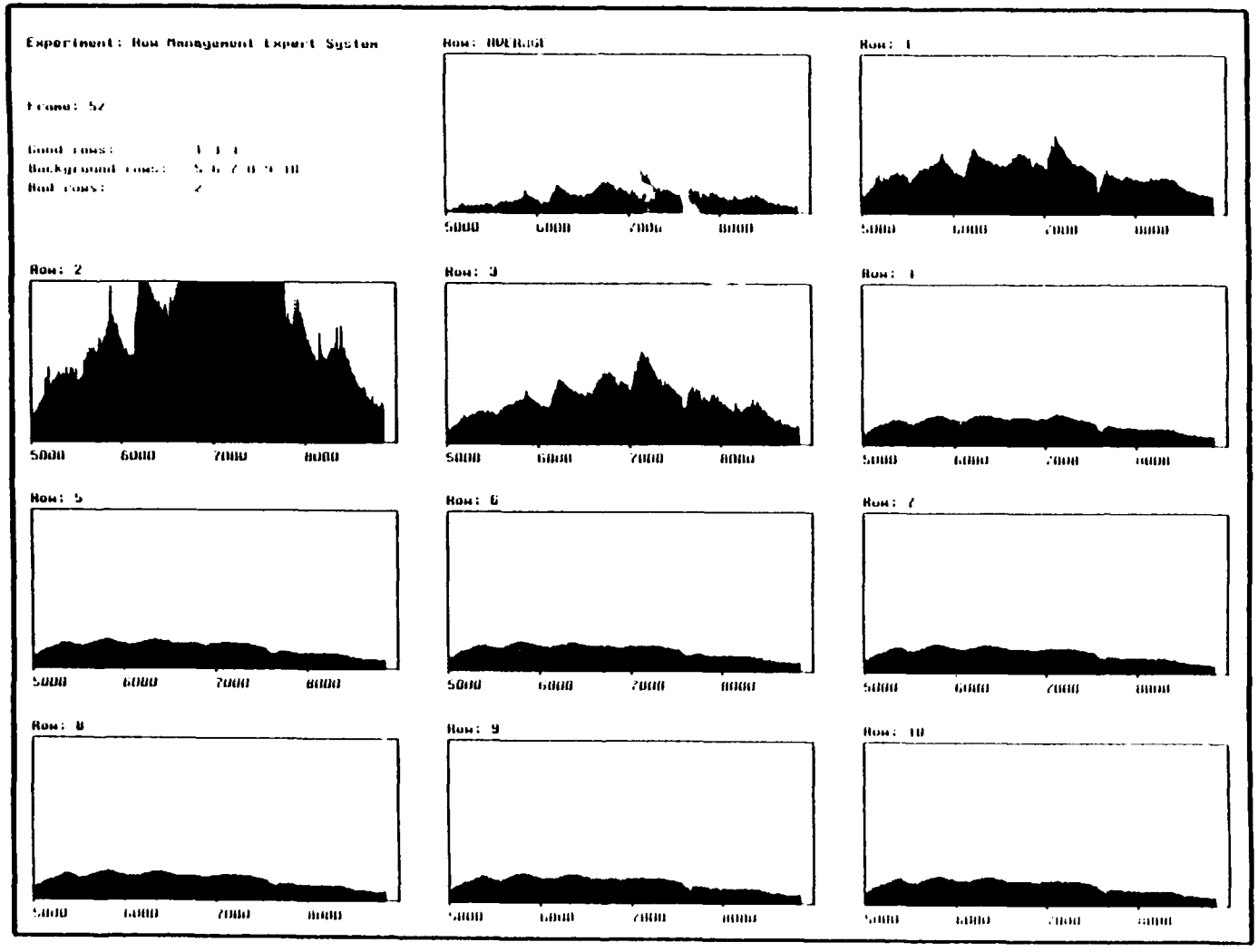

Figure 7. Row Management Example Frame

```
(deffacts average-pixel
     (pixel row  1  63)
     (pixel row  2 172)
     (pixel row  3  70)
     (pixel row  4  35)
     (pixel row  5  31)
     (pixel row  6  31)
     (pixel row  7  31)
     (pixel row  8  30)
     (pixel row  9  30)
     (pixel row 10  30) )

(deffacts area-deviation
     (dev row  1  84)
     (dev row  2 232)
     (dev row  3 112)
     (dev row  4  13)
     (dev row  5   3)
     (dev row  6   2)
     (dev row  7   1)
     (dev row  8   1)
     (dev row  9   1)
     (dev row 10   0) )

(deffacts bad-data
     (maxed row  2) )
```

An actual run of the CLIPS rule-base using the fact-list from the frame in Figure 7 and showing the firing sequence of the rules follows. The PRINT-ROW-DISPOSITIONS rule firings were deleted since they serve only to display the resulting fact-list.

```
==> f-0     (initial-fact)
==> f-1     (pixel row 1 63)
==> f-2     (pixel row 2 172)
==> f-3     (pixel row 3 70)
==> f-4     (pixel row 4 35)
==> f-5     (pixel row 5 31)
==> f-6     (pixel row 6 31)
==> f-7     (pixel row 7 31)
==> f-8     (pixel row 8 30)
==> f-9     (pixel row 9 30)
==> f-10    (pixel row 10 30)
==> f-11    (dev row 1 84)
==> f-12    (dev row 2 232)
==> f-13    (dev row 3 112)
==> f-14    (dev row 4 13)
```

```
==> f-15    (dev row 5 3)
==> f-16    (dev row 6 2)
==> f-17    (dev row 7 1)
==> f-18    (dev row 8 1)
==> f-19    (dev row 9 1)
==> f-20    (dev row 10 0)
==> f-21    (maxed row 2)
FIRE    1 saturated-row: f-21
==> f-22    (row 2 saturated)
FIRE    2 background-1: f-20,,
==> f-23    (row 10 background)
FIRE    3 background-1: f-19,,
==> f-24    (row 9 background)
FIRE    4 background-1: f-18,,
==> f-25    (row 8 background)
FIRE    5 background-1: f-17,,
==> f-26    (row 7 background)
FIRE    6 background-1: f-16,,
==> f-27    (row 6 background)
FIRE    7 background-1: f-15,,
==> f-28    (row 5 background)
FIRE    8 data-2: f-14,f-4,,,f-15,f-5,f-13,f-3
==> f-29    (row 4 data)
FIRE    9 data-2: f-14,f-4,,,f-15,f-5,f-13,f-3
FIRE   10 data-1: f-13,,
==> f-30    (row 3 data)
FIRE   11 data-1: f-11,,
==> f-31    (row 1 data)
```

The resulting facts from running the CLIPS rule-base using the fact-list from the frame in Figure 7 follows.

```
(row 1 data)
(row 3 data)
(row 4 data)
(row 5 background)
(row 6 background)
(row 7 background)
(row 8 background)
(row 9 background)
(row 10 background)
(row 2 saturated)
```

Note that Figure 1 presents the same frame as Figure 7 except that in Figure 1 a human analyst made the determination of the row dispositions,

while in Figure 7 the row management rule-base made the row disposition determination. The difference in the two determinations is the result of the human analyst missing the small TiO band just forming in row 4. This is a good example of how the expert system may be able to perform better than a human analyst.

This rule-base will not be able to make the correct determination of the row dispositions all the time, but it will allow for the easy addition of more rules which may become necessary in order to take care of special cases which may arise in the future. Appendix B presents a set of frames which represent a good cross section of the types of rows which are currently acquired. Included in this appendix is a figure of each frame along with the fact-list derived from that frame and the resulting facts after running the row management rule-base.

Spectra Wavelength Alignment

A major problem which required a great deal of attention was the alignment of the spectra. Because of the method of field measurements used to record the spectra, the pixel wavelength assignment is frequently inaccurate. This problem was handled by assuming that the wavelength bias was constant for a single sensor during a single mission (a good assumption since the duration of experiments is usually only one to two minutes).

The spectra were aligned by finding distinct spikes in the spectrum near where the Na, Li, and K lines are normally found and then calculating an average bias from these spikes. This average bias was

then applied to each of the frames of data in the set of spectra. Only 1% of the good rows must have a distinct spike for this technique to work correctly. While this procedure works well to align the spectra to the proper wavelength scale, it cannot function as a recognition scheme for atomic lines since it finds only pixels that are significantly different from the surrounding pixels. Figure 8 presents a 3-D representation of the aligned spectra from a segment of a single experiment. Note the alignment of the atomic species (Na and K) which are present.

Atomic Feature Extraction and Topology

The first neural network topology that was considered for recognition of atomic species had an input for each of the pixels with an output for each of the species which the neural network was trained to recognize. These inputs were to be fed through a two hidden layer neural network and the species present in the spectrum would have corresponded to the high firing neurons of the output layer. This neural network was eliminated as being too complex since several neurons would have to fire at once for the simultaneous recognition of multiple species within a row.

A second neural network topology considered was one that again would have an input for each of the pixels, but would have only two output nodes: one indicating that a species is present, and one indicating that a species is not present. Again, this topology was eliminated as having too many inputs to be efficient. Since there would have to be a separate neural network for each species, the number of

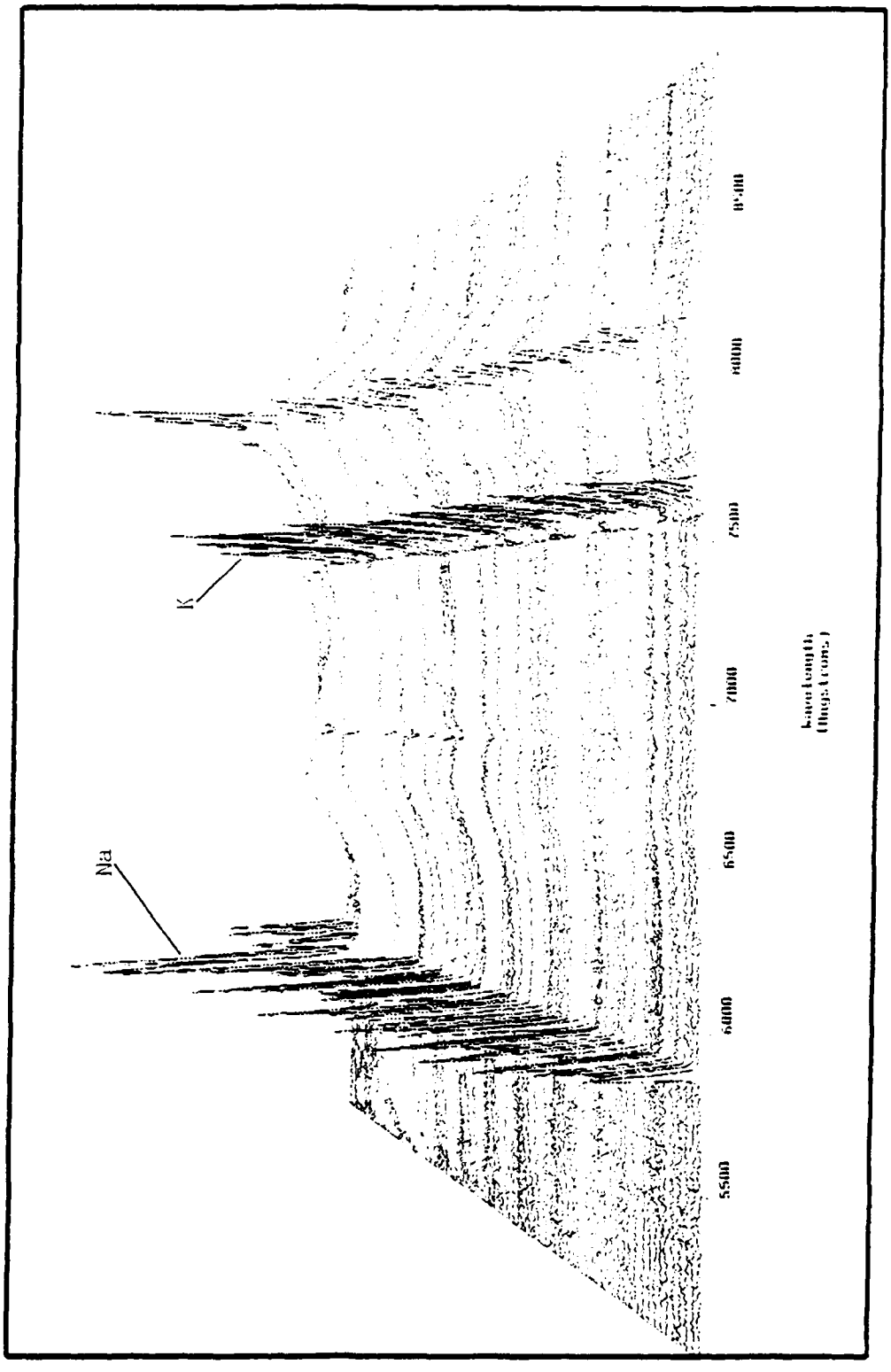

Figure 8. Aligned Spectra

inputs would have increased by 490 each time a new species was added to the recognizer.

A third topology that was considered utilized an averaging technique to combine the 490 pixels into a more manageable number. This was not considered to be an acceptable solution, at least for the atomic species, because there are a number of species that show up in only one or two pixels. It was felt that an average of the pixels might obscure these species, therefore this topology was also rejected.

The topology which was finally chosen takes, as the input vector, only a region around the pixels where a species is normally seen. The output layer contains two nodes, one for indicating a species is present and one for indicating a species is not present. This reduces the number of pixels used as input for the neural network to around ten to twenty, a more manageable number. It also provides the capability to have several different neural networks, one for each species, which makes the recognition of each species more efficient. For the atomic species (Li, Na, and K) the neural network consisted of a nine component input vector, with the first hidden layer containing six nodes and the second hidden layer containing three nodes. The output layer contained two output nodes.

Molecular Feature Extraction and Topology

Once the development of neural networks for recognition of atomic species was completed, the problem of recognizing molecular species was addressed. This area is more difficult because the species appear as bands in the spectrum and therefore more inputs are needed for the

neural network to be adequately trained. In order to use the same general approach that was taken in the development of the atomic recognizer, only a few neural network topologies were considered.

The primary problem that had to be solved was how to represent the essence of the molecular species in a reasonable size vector. One idea, as already described in the case of atomic species, was to take the large wavelength range of the molecular species and combine them into a more manageable number. This might have worked, but since the system already existed to take the raw pixels around a species, it was decided to try to use the same feature set extraction approach that was applied to the atomic species.

In order to keep the number of pixels to a minimum, an approach was chosen which used only a region around the molecular band which appears the most intensely in the spectrum. This means that a range of about forty pixels is required for the broadest molecular species. Many bands of molecular species have a distinct rise and fall; some are abrupt while others are gradual. By focusing the pixel range in on one edge of the band, the number of pixels was reduced even further, from forty down to twenty. This greatly reduces the number of calculations which the neural network has to perform and therefore increases the recognition speed, with a minimal degradation in the accuracy of the neural network.

Vector Normalization

The data vectors had to be normalized before they were fed to the neural network. The normalization procedure used was to take each pixel

(input to the neural network) and divide it by a normalization factor for that specific pixel number. The formula used for normalizing was:

$$N(\epsilon_{\lambda\iota}) = \frac{\epsilon_{\lambda\iota} - \bar{X}_{\lambda}}{\sigma_{\lambda}} \tag{7}$$

where

$N(\epsilon_{\lambda\iota})$ = Normalized value for pixel λ of exemplar ι
$\epsilon_{\lambda\iota}$ = Value for pixel λ of exemplar ι
$\bar{X}_{\lambda}$ = Mean of pixel λ values for training set
σ_{λ} = Variance of pixel λ values for training set
λ = Exemplar vector component number
ι = Exemplar number

Summary

Figure 9 shows a hierarchical summary of the processes used in this project to make species identifications in raw spectral data. The five basic tasks (row management, wavelength alignment, feature extraction, vector normalization, and neural network recognition) are shown in relation to one other. This provides the basic groundwork and knowledge for the results which are presented in the next chapter.

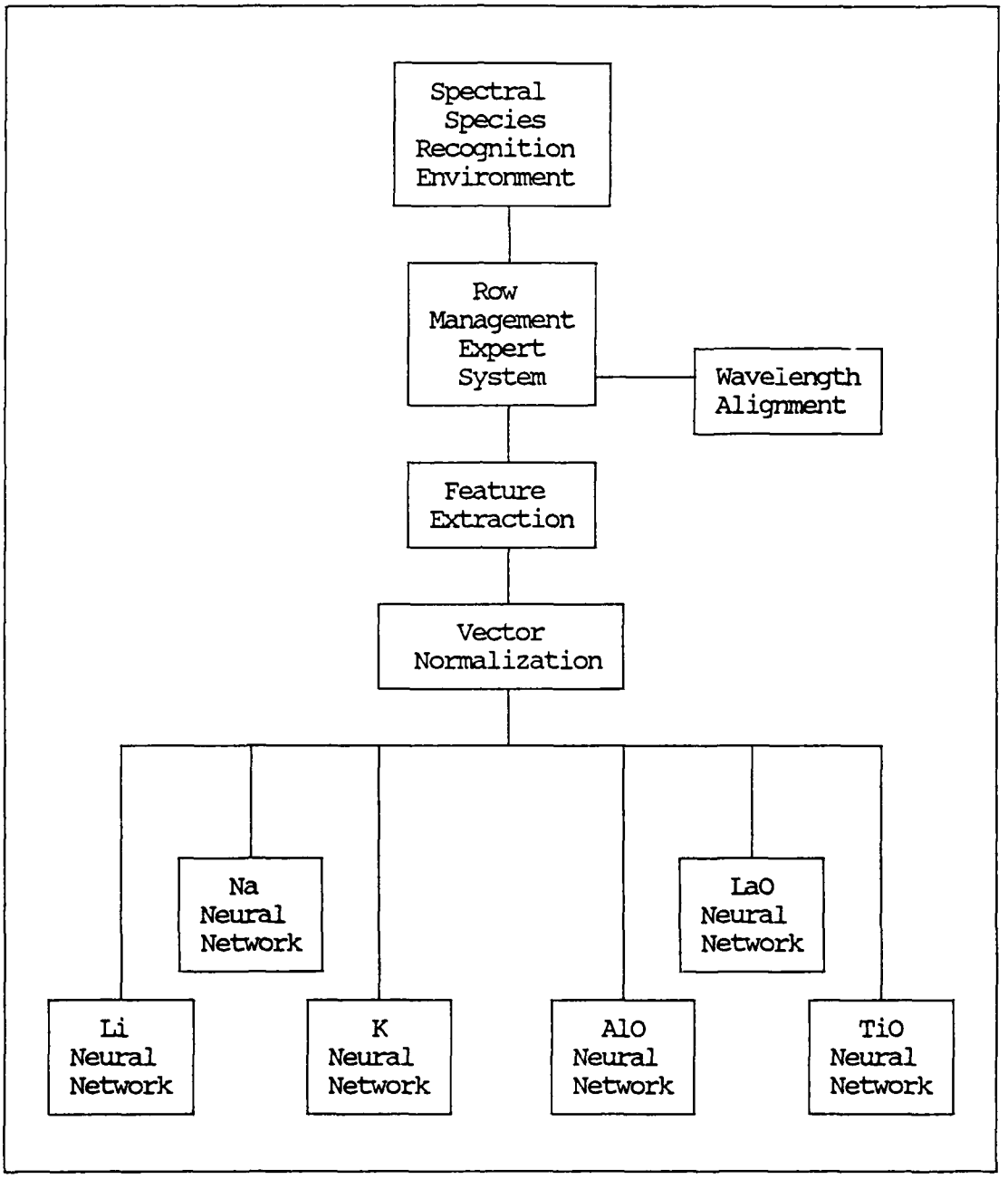

Figure 9. Current Species Recognition Process Hierarchy

IV. Results

Atomic Species

Initially, a neural network for K was trained with data that had not been corrected for the misalignment of the spectra. The performance of this neural network was not acceptable, probably due to to the vast differences between sensors as to where the species was located. Once the alignment problem was solved, a neural network for K was trained with twenty input nodes (ten pixels on each side of the wavelength where K is most prominent), ten nodes in the first hidden layer, five nodes in the second hidden layer, and two output nodes. Several neural networks for the recognition of K were trained, each time with the number of input nodes and the number of hidden nodes decreasing. Once the neural network configuration for K was finalized, the same configuration was tried for Li and Na to reduce development time. Since this configuration proved to be acceptable for all atomic species, no further development was necessary. The wavelength ranges from which the exemplars for the atomic species neural networks were obtained are presented in Table I.

The final neural network configurations and training parameters used for the atomic species (Li, Na, and K) are presented in Table II. Two general rules were followed in the development of training parameters to prevent memorization by the neural network: that the number of exemplars be greater than three times the test vector length, and that the number of exemplars divided by the number of weights be greater than two. There was no significant decrease in the performance

Table I

Atomic Species Exemplar Sources

Species	Li	Na	K
Species Position in Spectrum (μm)	.6708	.5890	.7699
Lower Wavelength Bound (μm)	.6676	.5858	.7667
Upper Wavelength Bound (μm)	.6740	.5922	.7731
Wavelength Range (μm)	.0064	.0064	.0064

Table II

Configuration and Parameters of Atomic Species Neural Networks

Species	Li	Na	K
Input Vector Length	9	9	9
# Nodes in 1st Hidden Layer	6	6	6
# Nodes in 2nd Hidden Layer	3	3	3
Output Vector Length	2	2	2
Weight Paths	324	324	324
# Training Exemplars (Rows)	877	875	1333
# Test Exemplars (Rows)	877	875	1333
# of Frames (Total)	481	482	716
# Training Iterations	100000	100000	100000

of these neural networks as compared to the neural networks which were trained with larger input vectors and more weight paths.

Each atomic species neural network was trained ten different times so that there were ten sets of weights which could be used by a specific species neural network. Each of the ten training sessions started with a random weight seed. The statistical results of these different weight sets for each of the atomic species are presented in Table III. Each time the neural network was trained, the original exemplar set was randomly divided into two exemplar sets of equal size so that the statistical results would be cross-validated (19:111-133). The first set was used as the training exemplars while the second set was used as the test exemplars. The exemplars were taken from both a nighttime experiment and a daytime experiment, so that the effects of the different backgrounds could be learned by the neural network. The training and test sets of exemplars have a random number of daytime and nighttime exemplars.

As can be seen from the results in Table III, the recognition rates for the atomic species look promising. The confidence intervals from the statistical results of the atomic species neural networks are presented in Table IV. The use of a cross-validation technique to acquire the statistical results and the fact that the confidence intervals even at the 99% level are small, at most only about four percent, is an assurance that the performance of the neural networks is not an anomaly of the data. It should be noted that the neural networks trained up to the given recognition rates in about 20,000 iterations for Na and Li, and 40,000 iterations for K. The slow training time as well

Table III

Atomic Species Neural Network Recognition Rates

Species	Li		Na		K	
Exemplar Set	Train	Test	Train	Test	Train	Test
Sample Size	10	10	10	10	10	10
Mean (%)	**84.98**	**84.56**	**88.77**	**91.11**	**80.13**	**80.84**
Median (%)	**85.02**	**85.01**	**88.84**	**91.20**	**80.26**	**79.82**
Minimum (%)	84.71	79.82	88.40	89.26	79.05	79.52
Maximum (%)	85.21	86.89	89.05	92.11	80.90	85.52
Range (%)	.50	7.07	.65	2.85	1.85	6.00
Standard Deviation (%)	.192	2.028	.265	.981	.624	1.996
Variance (%)	.037	4.112	.070	.963	.389	3.984

Table IV

Confidence Intervals for Atomic Species Neural Networks

Species	Li		Na		K	
Exemplar Set	Train	Test	Train	Test	Train	Test
90% Lower Bound	84.87	83.39	88.62	90.54	79.77	79.68
90% Upper Bound	85.09	85.74	88.93	91.68	80.49	82.00
95% Lower Bound	84.84	83.11	88.58	90.41	79.68	79.41
95% Upper Bound	85.12	86.01	88.96	91.81	80.57	82.27
99% Lower Bound	84.78	82.48	88.50	90.10	79.49	78.79
99% Upper Bound	85.18	86.65	89.05	92.12	80.77	82.89

as the lower recognition rates achieved in the recognition of K are probably a result of a molecular oxygen (O_2) absorption band that occurs near where K radiates most prominently. (See Figure 4 for the relative positions of K and O_2.)

Molecular Species

When neural networks were being developed for the molecular species, the results were very promising right from the beginning. Because of this, the number of pixels and weight paths were drastically reduced as discussed previously in order to decrease training time as well as to prevent the neural networks from memorizing the exemplar sets. Table V presents the wavelength ranges from which the exemplars for the molecular species neural networks were obtained.

Table V

Molecular Species Exemplar Sources

Species	AlO	LaO	TiO
Species Band Wavelength - Lower (μm)	.5337	.7380	.7054
Species Band Wavelength - Upper (μm)	.5423	.7465	.7269
Lower Wavelength Bound (μm)	.5250	.7274	.6994
Upper Wavelength Bound (μm)	.5402	.7426	.7146
Wavelength Range (μm)	.0152	.0152	.0152

The initial neural network topology used for the molecular species included forty pixels in the input vector, twenty nodes in the first

hidden layer, ten nodes in the second hidden layer, and two output nodes. As discussed previously, the molecular neural network topologies were altered to take advantage of a priori knowledge. The final neural network configurations and training parameters used for the molecular species (AlO, LaO, and TiO) are presented in Table VI. The reason that there are fewer training exemplars for AlO compared to the number of exemplars for the other species, is that AlO does not show up in spectra as often as the other species making it difficult to find exemplars for this species.

Table VI

Configuration and Parameters of Molecular Species Neural Networks

Species	AlO	LaO	TiO
Input Vector Length	20	20	20
# Nodes in 1st Hidden Layer	10	10	10
# Nodes in 2nd Hidden Layer	5	5	5
Output Vector Length	2	2	2
Weight Paths	2000	2000	2000
# Training Exemplars (Rows)	640	876	863
# Test Exemplars (Rows)	640	876	863
# of Frames (Total)	397	480	462
# Training Iterations	100000	100000	100000

The general rule that the number of exemplars be greater than three times the test vector length was followed. However, the rule that the

number of exemplars divided by the number of weights be greater than two to prevent memorization by the neural network was not achieved. This was due to the fact that the recognition rate of a molecular species neural network with fewer weight paths was not as good as the final neural network for a reasonable number of training iterations. Even though this second condition was not met, the results were such that there seemed to be no memorization by the neural networks. There was a small, only two or three percent, but insignificant decrease in the performance of these neural networks as compared to the neural networks which were trained with larger input vectors and more weight paths.

Each molecular species neural network was trained ten different times so that there were ten sets of weights for each species neural network. Each of the ten training sessions started with a random weight seed. The statistical results of these different weight sets for each of the molecular species are presented in Table VII. The procedures for training with the molecular species were the same as those used for training with the atomic species. Each time the neural network was trained, the original exemplar set was randomly divided into two exemplar sets of equal size so that the statistical results would be cross-validated (19:111-133). The first set was used as the training exemplars while the second set was used as the test exemplars. The exemplars were taken from both a nighttime experiment and a daytime experiment, so that the effects of the different backgrounds could be learned by the neural network. The training and test sets of exemplars have a random number of daytime and nighttime exemplars.

Table VII

Molecular Species Neural Network Recognition Rates

Species	AlO		LaO		TiO	
Exemplar Set	Train	Test	Train	Test	Train	Test
Sample Size	10	10	10	10	10	10
Mean (%)	**87.03**	**89.48**	**86.62**	**92.95**	**91.76**	**88.99**
Median (%)	**88.09**	**89.38**	**86.59**	**94.75**	**91.88**	**89.86**
Minimum (%)	80.36	86.09	86.30	85.16	91.10	83.20
Maximum (%)	89.36	93.13	87.14	95.55	92.45	91.77
Range (%)	9.00	7.04	.84	10.39	1.35	8.57
Standard Deviation (%)	2.736	2.539	.309	3.615	.436	3.101
Variance (%)	7.484	6.444	.095	13.071	.190	9.615

As can be seen from the results in Table VII, the recognition rates for the molecular species look just as promising as those for the atomic species. The confidence intervals from the statistical results of the molecular species neural networks are presented in Table VIII. The use of a cross-validation technique to acquire the statistical results and the fact that the confidence intervals even at the 99% level are small, at most only seven percent, is an assurance that the performance of the neural networks is not an anomaly of the data. It should be noted that the neural networks trained up to the given recognition rates in about 40,000 iterations for all of the molecular species.

Table VIII

Confidence Intervals for Molecular Species Neural Networks

Species	AlO		LaO		TiO	
Exemplar Set	Train	Test	Train	Test	Train	Test
90% Lower Bound	85.44	88.01	86.44	90.85	91.51	87.20
90% Upper Bound	88.62	90.96	86.80	95.04	92.02	90.79
95% Lower Bound	85.07	87.67	86.40	90.36	91.45	86.77
95% Upper Bound	88.99	91.30	86.84	95.53	92.08	91.21
99% Lower Bound	84.22	86.87	86.30	89.23	91.32	85.81
99% Upper Bound	89.84	92.09	86.94	96.66	92.21	92.18

Atomic Species Within Frames

The fact that for a species to be correctly identified in a frame, it need only be identified in one of the rows of that frame opens the possibility of increasing the recognition rate of the neural networks. Table IX presents the parameters for the neural networks which were trained for use in determining the frame recognition rates for atomic species. The configurations of the neural networks used were the same as those which were used in the original atomic species neural networks presented earlier.

Each atomic species frame recognition neural network was trained ten different times, each time beginning with a random weight seed, just as with the row only training. Table X presents the increased recognition rates for atomic species as a result of looking at a

Table IX

Parameters of Frame Recognition Atomic Species Neural Networks

Species	Li	Na	K
# Training Exemplars (Rows)	931	931	1660
# Test Exemplars (Rows)	824	819	1006
# Training Frames	228	228	453
# Test Frames	253	254	263
# Training Iterations	100000	100000	100000

spectral frame as a whole, not as just a single row. If a species is present in a frame and is found in at least one row of that frame, then the frame is correct. However, if a species is not present in a frame, but the neural network identifies it as being present, the entire frame is incorrect. Note the average increase in the mean and median recognition rates for each of the atomic species using both the training and test exemplar sets.

The neural networks were trained on sequential sets of spectral data just as they are found in the raw data. The Li and Na neural networks were trained solely on daytime data and were tested on nighttime data while the the K neural networks were trained on both daytime and nighttime data but tested with only nighttime data. The data could not be randomly mixed as with the previous atomic neural networks discussed since the exemplars must be kept in row order to have any meaning when discussing overall frame recognition rates.

Table X

Atomic Species Neural Network Frame Recognition Rates

Species	Li		Na		K	
Exemplar Set	Train	Test	Train	Test	Train	Test
Sample Size	10	10	10	10	10	10
R Mean (%)	83.06	80.91	94.04	86.48	79.31	80.98
F Mean (%)	96.14	95.26	98.07	97.68	94.46	97.61
Δ Mean (%)	**13.08**	**14.35**	**4.03**	**11.20**	**15.15**	**16.63**
R Median (%)	83.06	80.46	94.04	86.50	79.16	81.71
F Median (%)	96.93	95.85	97.81	97.84	94.59	97.72
Δ Median (%)	**13.87**	**15.39**	**3.77**	**11.34**	**15.43**	**16.01**
R Minimum (%)	82.79	78.52	93.61	83.39	78.83	75.25
F Minimum (%)	91.67	92.09	97.81	96.06	91.61	95.06
R Maximum (%)	83.33	86.29	94.56	89.38	80.50	87.67
F Maximum (%)	97.81	97.63	98.68	98.82	97.57	98.86
R Range (%)	.54	7.77	.95	5.99	1.67	12.42
F Range (%)	6.14	5.54	.87	2.76	5.96	3.80
R Standard Deviation (%)	.196	2.616	.366	2.065	.509	3.989
F Standard Deviation (%)	2.206	2.245	.367	.799	1.779	1.030
R Variance (%)	.038	6.845	.134	4.263	.260	15.910
F Variance (%)	4.867	5.040	.135	.638	3.166	1.061

R = Row
F = Frame
Δ = Increase in Frame over Row

As is shown in Table X, the row recognition rates from the exemplars are significantly different from those presented earlier. This is due to the fact that the neural networks in Table X were trained with data that were not randomly distributed among the training and test sets. Switching the training sets for the test sets and the test sets for the training sets was attempted; however these neural networks tended to do poorly since there was less noise in nighttime data compared to the daytime data for the neural network to learn around. The confidence intervals for the data presented in Table X are presented in Table XI. Again, these intervals give some assurance that the results presented are not an anomaly of the data.

Molecular Species Within Frames

The increase in the recognition rates by using the correct frame criteria was also successful for the molecular species. Table XII presents the parameters for the neural networks which were trained for use in determining the frame recognition rates for molecular species. The configurations of the neural networks used were the same as those which were used in the original molecular species neural networks presented earlier.

Each molecular species frame recognition neural network was trained ten different times, each time beginning with a random weight seed, just as with the row only training. Table XIII presents the increased recognition rates for molecular species as a result of looking at a spectral frame as a whole, not as just a single row. If a species is present in a frame and is found in at least one row of that frame, then

Table XI

Confidence Intervals for Frame Recognition Atomic Species Neural Networks

Species	Li		Na		K	
Exemplar Set	Train	Test	Train	Test	Train	Test
R 90% Lower Bound	82.94	79.39	93.83	85.28	79.01	78.67
R 90% Upper Bound	83.17	82.43	94.25	87.68	79.60	83.30
F 90% Lower Bound	94.86	93.96	97.86	97.21	93.43	97.01
F 90% Upper Bound	97.42	96.56	98.28	98.14	95.49	98.20
R 95% Lower Bound	82.92	79.04	93.78	85.00	78.94	78.13
R 95% Upper Bound	83.20	82.78	94.30	87.96	79.67	83.84
F 95% Lower Bound	94.56	93.65	97.81	97.11	93.18	96.87
F 95% Upper Bound	97.72	96.86	98.33	98.25	95.73	98.34
R 99% Lower Bound	82.86	78.22	93.67	84.36	78.78	76.88
R 99% Upper Bound	83.26	83.60	94.42	88.60	79.83	85.08
F 99% Lower Bound	93.87	92.95	97.69	96.86	92.63	96.55
F 99% Upper Bound	98.41	97.56	98.45	98.50	96.29	98.66

R = Row
F = Frame

the frame is correct. However, if a species is not present in a frame, but the neural network identifies it as being present, the entire frame is incorrect. Note the average increase in the mean and median recognition rates for each of the molecular species using both the training and test exemplar sets. The large standard deviation and the

Table XII

Parameters of Frame Recognition Molecular Species Neural Networks

Species	AlO	LaO	TiO
# Training Exemplars (Rows)	880	933	931
# Test Exemplars (Rows)	401	819	796
# Training Frames	254	228	228
# Test Frames	143	252	234
# Training Iterations	100000	100000	100000

small increase in the test recognition rates for AlO are probably due to the small number of frames from which the exemplars were extracted.

The neural networks were trained on sequential sets of spectral data just as they are found in the raw data. The LaO and TiO neural networks were trained solely on daytime data and tested on nighttime data while the AlO neural networks were trained solely on nighttime data and tested on daytime data. AlO was trained on nighttime data because there were more sequential frames of nighttime data than daytime data. The data could not be randomly mixed as with the previous molecular neural networks discussed since the exemplars must be kept in row order to have any meaning when discussing overall frame recognition rates.

As is shown in Table XIII, the row recognition rates from the exemplars are significantly different from those presented earlier. This is due to the fact that the neural networks in Table XIII were trained with data that were not randomly distributed among the training

Table XIII

Molecular Species Neural Network Frame Recognition Rates

Species	AlO		LaO		TiO	
Exemplar Set	Train	Test	Train	Test	Train	Test
Sample Size	10	10	10	10	10	10
R Mean (%)	93.84	88.43	90.85	68.14	97.91	77.26
F Mean (%)	99.57	90.28	98.03	93.61	100.00	94.61
Δ Mean (%)	**5.73**	**1.85**	**7.18**	**25.47**	**2.09**	**17.35**
R Median (%)	93.99	87.90	90.92	67.46	97.89	76.76
F Median (%)	99.80	92.31	97.59	93.65	100.00	93.80
Δ Median (%)	**5.81**	**4.41**	**6.67**	**26.19**	**2.11**	**17.04**
R Minimum (%)	92.68	81.55	89.82	60.32	97.43	75.50
F Minimum (%)	98.82	80.42	96.93	90.87	100.00	91.88
R Maximum (%)	94.64	93.27	91.61	78.02	98.29	80.28
F Maximum (%)	100.00	97.90	100.00	96.83	100.00	97.86
R Range (%)	1.96	11.72	1.79	17.70	.86	4.78
F Range (%)	1.18	17.48	3.07	5.96	.00	5.98
R Standard Deviation (%)	.598	3.544	.589	4.991	.240	1.671
F Standard Deviation (%)	.539	6.886	1.155	2.387	.000	2.409
R Variance (%)	.357	12.562	.347	24.912	.058	2.794
F Variance (%)	.291	47.417	1.334	5.700	.000	5.803

R = Row
F = Frame
Δ = Increase in Frame over Row

and test sets. Since the switching of the training sets for the test sets and the test sets for the training sets with the atomic species was unsuccessful, it was not attempted for the molecular species. The confidence intervals for the data presented in Table XIII are presented in Table XIV. Again, these intervals give some assurance that the results presented are not an anomaly of the data.

Summary

In reviewing areas in which the neural networks made incorrect identifications for both the atomic and molecular species, it was noted that the vast majority occur at the beginning or end of a data segment, where the species begin or finish radiating. This is the same place where human spectral analysts have trouble as well. One possibility to increase the recognition ability of the neural networks is to have them look into adjacent frames and see if a specific species is present in an adjacent frame before making a final decision. This is the way human analysts currently make difficult species identifications.

The results of using a set of neural networks for the recognition of both atomic and molecular species in a spectrum have proven that this method is a viable solution to the problem of analyzing large amounts of spectral data. With recognition accuracies of around 95% for species when taken together in a frame, the reduction in the time required to analyze spectral data should be significant. The potential to give analysts the capability to develop their own neural networks for new species which may be seen in the future is a major advantage of using neural networks to solve this problem. By using the methods detailed in

Table XIV

Confidence Intervals for Frame Recognition Molecular Species Neural Networks

Species	AlO		LaO		TiO	
Exemplar Set	Train	Test	Train	Test	Train	Test
R 90% Lower Bound	93.49	86.37	90.51	65.25	97.77	76.29
R 90% Upper Bound	94.18	90.49	91.19	71.04	98.05	78.23
F 90% Lower Bound	99.26	86.29	97.36	92.23	100.00	93.22
F 90% Upper Bound	99.88	94.27	98.70	95.00	100.00	96.01
R 95% Lower Bound	93.41	85.89	90.43	64.57	97.74	76.07
R 95% Upper Bound	94.26	90.97	91.27	71.72	98.08	78.46
F 95% Lower Bound	99.18	85.35	97.20	91.90	100.00	92.89
F 95% Upper Bound	99.95	95.21	98.85	95.32	100.00	96.34
R 99% Lower Bound	93.22	84.79	90.24	63.01	97.66	75.54
R 99% Upper Bound	94.45	92.07	91.45	73.27	98.16	78.98
F 99% Lower Bound	99.01	83.20	96.84	91.16	100.00	92.14
F 99% Upper Bound	100.12	97.36	99.21	96.06	100.00	97.09

R = Row
F = Frame

this thesis, these future species neural networks should have recognition rates at least as high as those already developed.

V. Conclusions and Recommendations

This research focused on the development of an integrated rule-based expert system and supervised artificial neural network with error correction learning. The objective was to develop a rule-base and neural network which would perform the preliminary analysis and save the analysts from the mundane task of reviewing thousands of spectral frames. The results presented in this thesis show that this goal was successfully achieved with the input to the neural network consisting of unclassified raw spectral data which had been previously filtered by the expert system, and the output consisting of the classification of both atomic and molecular species in the source.

The objective of this project was not to develop a system that points out exactly where spectral features lie in a frame of spectral data. Rather, the focus of this project was on developing a spectral analysis system which determines which frames in a sequence of thousands contain certain atomic and molecular species. While this aids analysts by cueing them into regions in the data where an interesting event might have taken place, there is a great deal more that can be accomplished to reduce the workload of the analyst.

So far, recognition of both atomic and molecular species has been successful. The next step is to train new neural networks to recognize additional molecular species. Once this has been accomplished, additional rules may be added to the row management rule-base to take care of new row characteristics and the neural network recognition technique can be applied to new species which appear in future

experiments. After simply being shown examples of new species, the neural networks will be able to train up and recognize these new species in very short order.

Future improvements which will aid the analyst include a system to automatically save to a database the time ranges in specific experiments where certain spectral species radiate. These species-time histories could then be recalled to make comparisons between different experiments. Another aid for analysts is an expert system which will take the species present, along with other data such as temperature and emissivity, and present a list of possible materials which may have made up the material which was radiating. There is also the possibility of adding a system to point out in the spectrum exactly where a specific species is radiating.

Other improvements can be made to increase the recognition accuracy of both atomic and molecular species. The addition of a background subtraction routine is desirable to reduce noise. Another possible improvement is to correct the raw spectra for the sensor response by subtracting out the dark current levels of the sensor. LOWTRAN 7 can be applied to the raw data in order to correct the spectra for atmospheric transmission. However, since the LOWTRAN correction is not always accurate and the weather data which is needed for LOWTRAN is many times only an estimate, this is not a highly recommended addition to the spectral analysis system.

Another expert system may be added to increase the recognition accuracy of spectral species. This expert system would supervise the neural networks and make decisions when there is confusion as to which

species are present, such as the time when species begin or finish radiating. This expert system may even make use of fuzzy logic in determining which rules should fire. These additions should make the task of spectral analysis of the common radiators a simple and fast process. This will leave the analysts with time to develop more advanced analysis techniques and look for obscure species in the spectral data.

Figure 10 shows a hierarchical summary of the processes which have been recommended for implementation in the spectral analysis system on the Silicon Graphics workstation in order to make species identifications in raw spectral data. The five basic tasks (row management, wavelength alignment, feature extraction, vector normalization, and neural network recognition) are shown in relation to each other. The recommended additions (materials expert system, species time history, background subtraction, sensor response correction, and neural network supervisor) are also shown to provide insight as to how each process relates to one another. The atmospheric transmission correction is also included in this summary, but is not recommended due to the limitations discussed previously. Once this spectral analysis system is integrated into a single common environment, the time savings and increased accuracy will be enormous.

Several significant problems had to be solved to get the the spectral analysis system's recognition rates to an acceptable level. The separation of the rows with data from the background and saturated rows was successfully achieved by using a rule-based expert system. The alignment of the spectra from a single experiment was performed by using

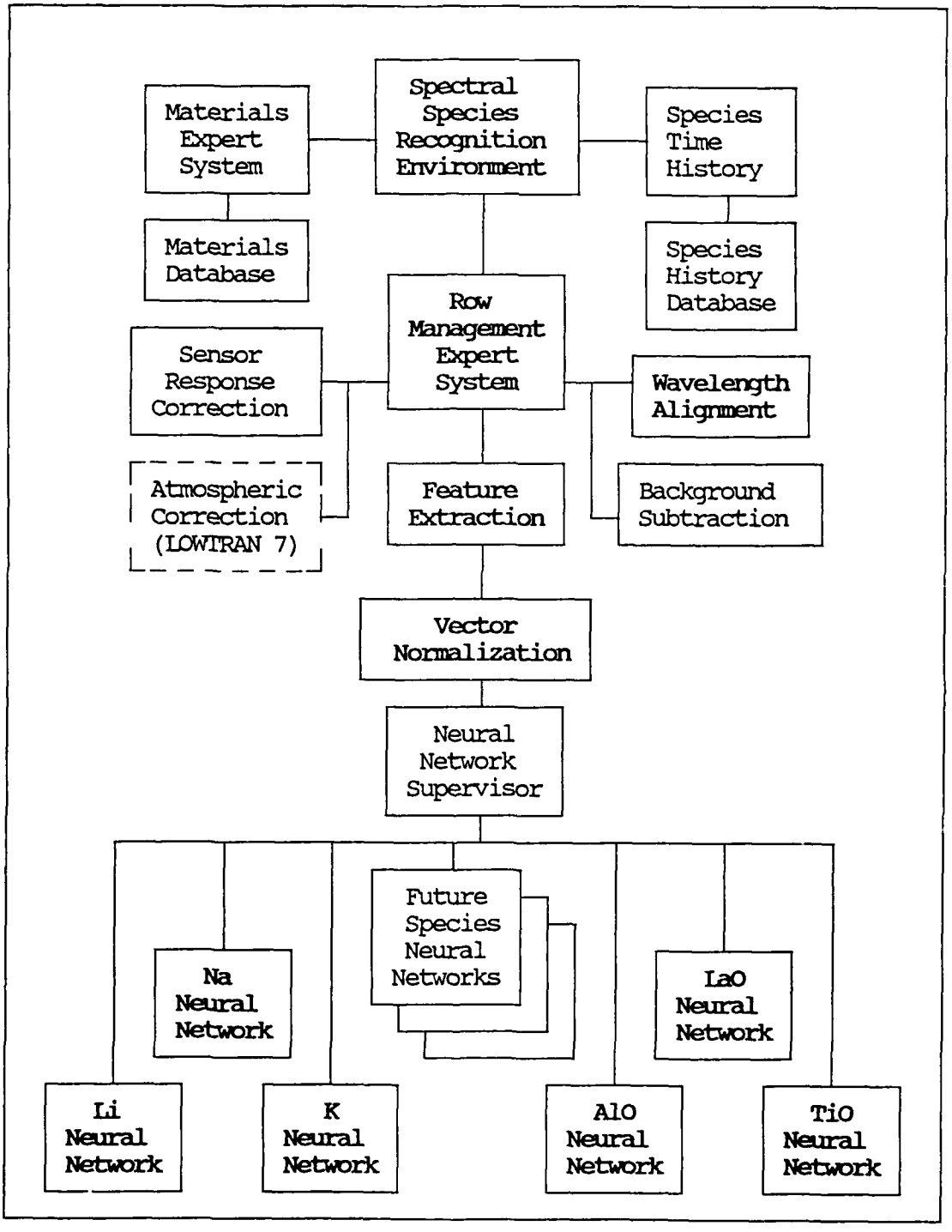

Figure 10. Future Species Recognition Process Hierarchy

an algorithmic approach. The actual selection of the feature space for the atomic species consisted of taking a range around the radiating line while for the molecular species it consisted of taking a range around one edge of the most prominent band. All of these developments contributed to a successful thesis which has provided a capability which will be used for the recognition of atomic and molecular species in visible spectra for years to come, and which may be applied to similar problems in order to make other analysis more efficient.

This thesis effort has been a major success and has made many significant contributions to the fields of spectral analysis, artificial intelligence, neural networks, and pattern recognition. The development of an integrated rule-based expert system and neural network is a significant accomplishment in the fields of artificial intelligence and neural networks. The use of this spectral analysis system for the identification of species in sets of multi-spectral data is a unique application that will increase the analysis capability of spectroscopists several times over. A reduction of two man-months of analysis per experiment will result from using this automated spectral recognition system and the accuracy of the identifications will increase. This cuts the time required to analyze an experiment almost in half. The contributions that have been made during the course of this thesis research span many disciplines and will have a lasting impact on spectral analysis, artificial intelligence, neural networks, and pattern recognition.

Appendix A: Row Management Rule-Base

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;                                                              ;;;
;;;                  ROW MANAGEMENT RULE-BASE                    ;;;
;;;                            FOR                               ;;;
;;; ATOMIC AND MOLECULAR SPECIES RECOGNITION NEURAL NETWORKS     ;;;
;;;                                                              ;;;
;;; This CLIPS rule-base uses a set of facts about a row to      ;;;
;;; determine the disposition of that row: data, background,     ;;;
;;; or saturated.  The disposition of a row is based on four     ;;;
;;; sets of facts which include:                                 ;;;
;;;   a) pixel - the average pixel count for a row               ;;;
;;;   b) dev   - the maximum deviation of a pixel from the       ;;;
;;;              row with the least area after smoothing         ;;;
;;;   c) maxed - a row which has six adjacent pixels which       ;;;
;;;              are at over 250 pixel counts                    ;;;
;;;   d) fall  - a row which has a sudden drop due to the        ;;;
;;;              saturation of the CCD                           ;;;
;;;                                                              ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;                                                              ;;;
;;; RULE - Data-1                                                ;;;
;;;                                                              ;;;
;;; This rule adds a DATA row fact to the fact list if any       ;;;
;;; of the rows have a deviation greater than or equal to 15.    ;;;
;;;                                                              ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defrule data-1
     (dev row ?number ?dev)
     (not (maxed row ?number))
        (not (fall row ?number))
     (test (>= ?dev 15))
        =>
        (assert (row ?number data)) )
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;                                                                    ;;;
;;; RULE - Data-2                                                      ;;;
;;;                                                                    ;;;
;;; This rule adds a DATA row fact to the fact list if any             ;;;
;;; of the interior rows which have a deviation between 10             ;;;
;;; and 15 have both adjacent rows with a deviation greater            ;;;
;;; than 10 or with a different average pixel count.                   ;;;
;;;                                                                    ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defrule data-2
     (dev row ?number1 ?dev1)
     (pixel row ?number1 ?pixel1)
     (not (maxed row ?number1))
     (not (fall row ?number1))
     (test (> ?dev1 10))
     (test (< ?dev1 15))
     (dev row ?number2 ?dev2)
     (pixel row ?number2 ?pixel2)
     (test (= (+ ?number1 1) ?number2))
     (or (test (> ?dev2 10))
         (test (!= ?pixel1 ?pixel2)) )
     (dev row ?number3 ?dev3)
     (pixel row ?number3 ?pixel3)
     (test (= (- ?number1 1) ?number3))
     (or (test (> ?dev3 10))
         (test (!= ?pixel1 ?pixel3)) )
     =>
     (assert (row ?number1 data)) )
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;                                                                    ;;;
;;; RULE - Data-3                                                      ;;;
;;;                                                                    ;;;
;;; This rule adds a DATA row fact to the fact list if row             ;;;
;;; 1 has a deviation between 10 and 15, and row 2 has a               ;;;
;;; deviation greater than 10 or a different average pixel             ;;;
;;; count.                                                             ;;;
;;;                                                                    ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defrule data-3
     (dev row 1 ?dev1)
     (pixel row 1 ?pixel1)
     (not (maxed row 1))
     (not (fall row 1))
     (test (> ?dev1 10))
     (test (< ?dev1 15))
     (dev row 2 ?dev2)
     (pixel row 2 ?pixel2)
     (or (test (> ?dev2 10))
         (test (!= ?pixel1 ?pixel2)) )
     =>
     (assert (row 1 data)) )

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;                                                                    ;;;
;;; RULE - Data-4                                                      ;;;
;;;                                                                    ;;;
;;; This rule adds a DATA row fact to the fact list if row             ;;;
;;; 10 has a deviation between 10 and 15, and row 9 has a              ;;;
;;; deviation greater than 10 or a different average pixel             ;;;
;;; count.                                                             ;;;
;;;                                                                    ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defrule data-4
     (dev row 10 ?dev1)
     (pixel row 10 ?pixel1)
     (not (maxed row 10))
     (not (fall row 10))
     (test (> ?dev1 10))
     (test (< ?dev1 15))
     (dev row 9 ?dev2)
     (pixel row 9 ?pixel2)
     (or (test (> ?dev2 10))
         (test (!= ?pixel1 ?pixel2)) )
     =>
     (assert (row 10 data)) )
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;                                                                    ;;;
;;; RULE - Background-1                                                ;;;
;;;                                                                    ;;;
;;; This rule adds a BACKGROUND row fact to the fact list if           ;;;
;;; any of the rows have a deviation less than or equal to             ;;;
;;; 10.                                                                ;;;
;;;                                                                    ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defrule background-1
        (dev row ?number ?dev)
          (not (maxed row ?number))
     (not (fall row ?number))
     (test (<= ?dev 10))
     =>
     (assert (row ?number background)) )

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;                                                                    ;;;
;;; RULE - Background-2                                                ;;;
;;;                                                                    ;;;
;;; This rule adds a BACKGROUND row fact to the fact list if           ;;;
;;; any of the rows which have a deviation between 10 and 15           ;;;
;;; have the adjacent row below (row + 1) with a deviation             ;;;
;;; less than or equal to 10, and with the same average pixel          ;;;
;;; count.                                                             ;;;
;;;                                                                    ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defrule background-2
     (dev row ?number1 ?dev1)
     (pixel row ?number1 ?pixel1)
     (not (maxed row ?number1))
     (not (fall row ?number1))
     (test (> ?dev1 10))
     (test (< ?dev1 15))
     (dev row ?number2 ?dev2)
     (pixel row ?number2 ?pixel2)
     (test (= (+ ?number1 1) ?number2))
     (test (<= ?dev2 10))
     (test (= ?pixel1 ?pixel2))
     =>
     (assert (row ?number1 background)) )
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;                                                                  ;;;
;;; RULE - Background-3                                              ;;;
;;;                                                                  ;;;
;;; This rule adds a BACKGROUND row fact to the fact list if         ;;;
;;; any of the rows which have a deviation between 10 and 15         ;;;
;;; have the adjacent row above (row - 1) with a deviation           ;;;
;;; less than or equal to 10, and with the same average pixel        ;;;
;;; count.                                                           ;;;
;;;                                                                  ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defrule background-3
     (dev row ?number1 ?dev1)
     (pixel row ?number1 ?pixel1)
     (not (maxed row ?number1))
     (not (fall row ?number1))
     (test (> ?dev1 10))
     (test (< ?dev1 15))
     (dev row ?number2 ?dev2)
     (pixel row ?number2 ?pixel2)
     (test (= (- ?number1 1) ?number2))
     (test (<= ?dev2 10))
     (test (= ?pixel1 ?pixel2))
     =>
     (assert (row ?number1 background)) )

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;                                                                  ;;;
;;; RULE - Saturated-Row                                             ;;;
;;;                                                                  ;;;
;;; This rule adds a SATURATED row fact to the fact list if          ;;;
;;; any row is maxed or has a fallen region.                         ;;;
;;;                                                                  ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defrule saturated-row
     (or (maxed row ?number)
         (fall row ?number) )
     =>
     (assert (row ?number saturated)) )
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;                                                                       ;;;
;;; RULE - Print-Row-Dispositions                                         ;;;
;;;                                                                       ;;;
;;; This rule prints out all the ROW facts once all the other rules       ;;;
;;; have fired.                                                           ;;;
;;;                                                                       ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defrule print-row-dispositions
     (declare (salience -1))
     (row ?number ?disposition)
     =>
     (fprintout t "(row " ?number " " ?disposition ")" crlf) )
```

Appendix B: Sample Fact-Lists with Frames

Frame 1

Facts presented to rule-base:

```
(deffacts average-pixel
     (pixel row  1   9)
     (pixel row  2   9)
     (pixel row  3   9)
     (pixel row  4   9)
     (pixel row  5   9)
     (pixel row  6   9)
     (pixel row  7   9)
     (pixel row  8   9)
     (pixel row  9   9)
     (pixel row 10   9) )

(deffacts area-deviation
     (dev row  1   2)
     (dev row  2   1)
     (dev row  3   2)
     (dev row  4   1)
     (dev row  5   2)
     (dev row  6   1)
     (dev row  7   1)
     (dev row  8   1)
     (dev row  9   2)
     (dev row 10   1) )
```

Resulting facts from running rule-base:

```
(row 1 background)
(row 2 background)
(row 3 background)
(row 4 background)
(row 5 background)
(row 6 background)
(row 7 background)
(row 8 background)
(row 9 background)
(row 10 background)
```

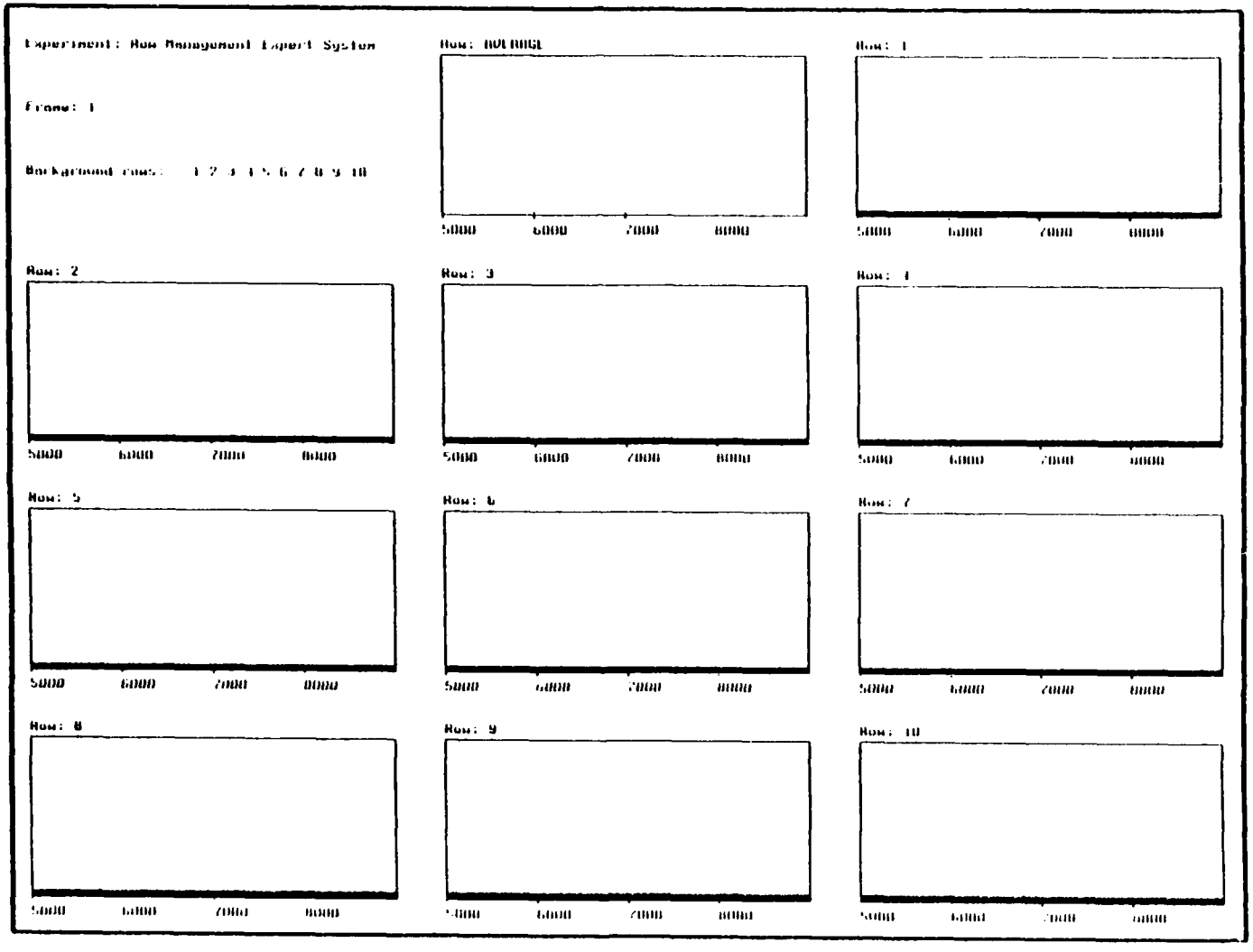

Figure 11. Row Management Frame 1

Frame 2

Facts presented to rule-base:

```
(deffacts average-pixel
     (pixel row  1   9)
     (pixel row  2   9)
     (pixel row  3   9)
     (pixel row  4   9)
     (pixel row  5   9)
     (pixel row  6  11)
     (pixel row  7  11)
     (pixel row  8  10)
     (pixel row  9   9)
     (pixel row 10   9) )

(deffacts area-deviation
     (dev row  1   2)
     (dev row  2   1)
     (dev row  3   1)
     (dev row  4   2)
     (dev row  5   8)
     (dev row  6  23)
     (dev row  7  20)
     (dev row  8  14)
     (dev row  9   4)
     (dev row 10   1) )
```

Resulting facts from running rule-base:

```
(row 1 background)
(row 2 background)
(row 3 background)
(row 4 background)
(row 5 background)
(row 6 data)
(row 7 data)
(row 8 data)
(row 9 background)
(row 10 background)
```

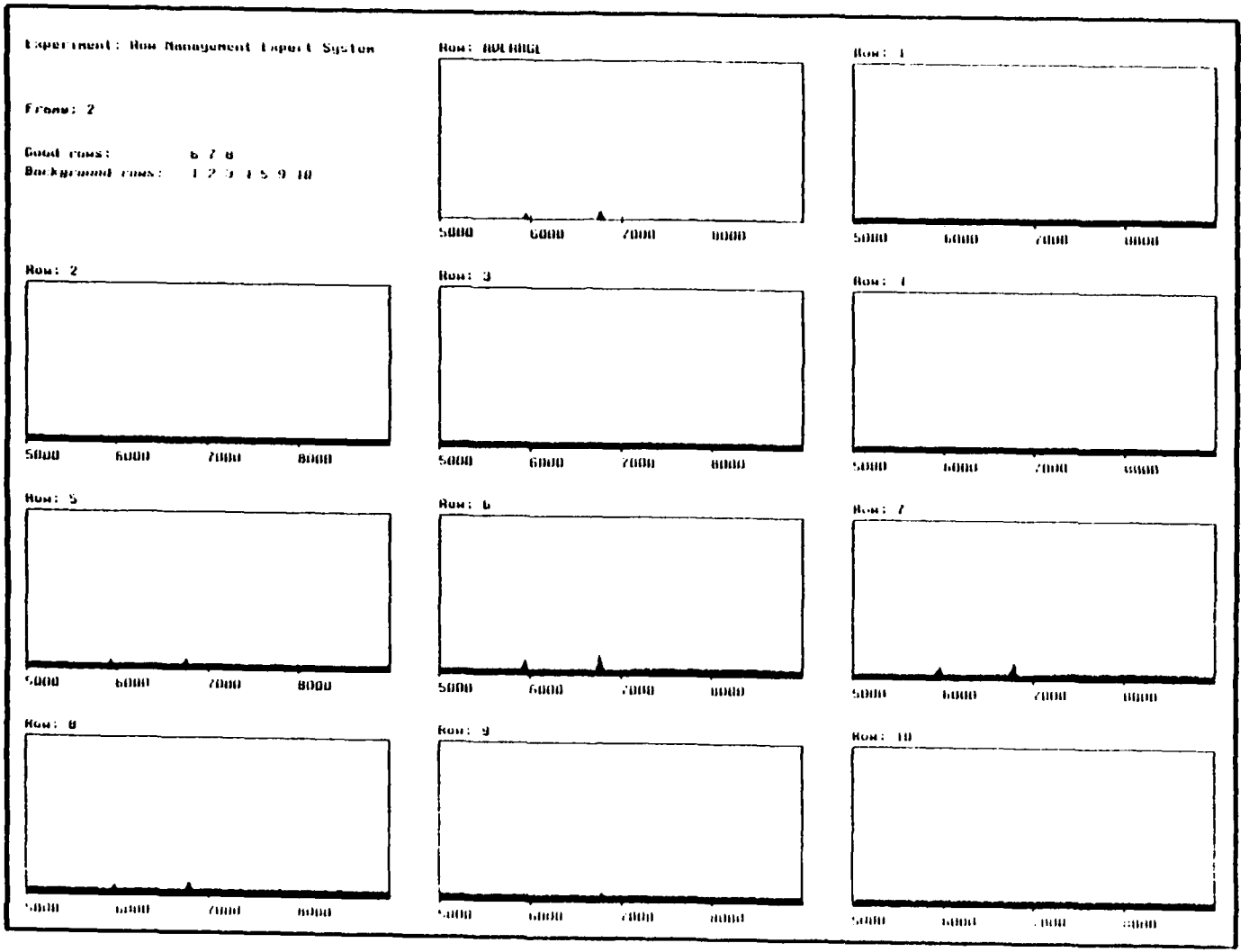

Figure 12. Row Management Frame 2

Frame 3

Facts presented to rule-base:

```
(deffacts average-pixel
     (pixel row  1   9)
     (pixel row  2   9)
     (pixel row  3  10)
     (pixel row  4  12)
     (pixel row  5  65)
     (pixel row  6  86)
     (pixel row  7  25)
     (pixel row  8  13)
     (pixel row  9  11)
     (pixel row 10  10) )

(deffacts area-deviation
     (dev row  1   0)
     (dev row  2   1)
     (dev row  3   2)
     (dev row  4   9)
     (dev row  5 180)
     (dev row  6 241)
     (dev row  7  59)
     (dev row  8  15)
     (dev row  9   7)
     (dev row 10   4) )

(deffacts bad-data
     (fall  row  6) )
```

Resulting facts from running rule-base:

```
(row 1 background)
(row 2 background)
(row 3 background)
(row 4 background)
(row 5 data)
(row 7 data)
(row 8 data)
(row 9 background)
(row 10 background)
(row 6 saturated)
```

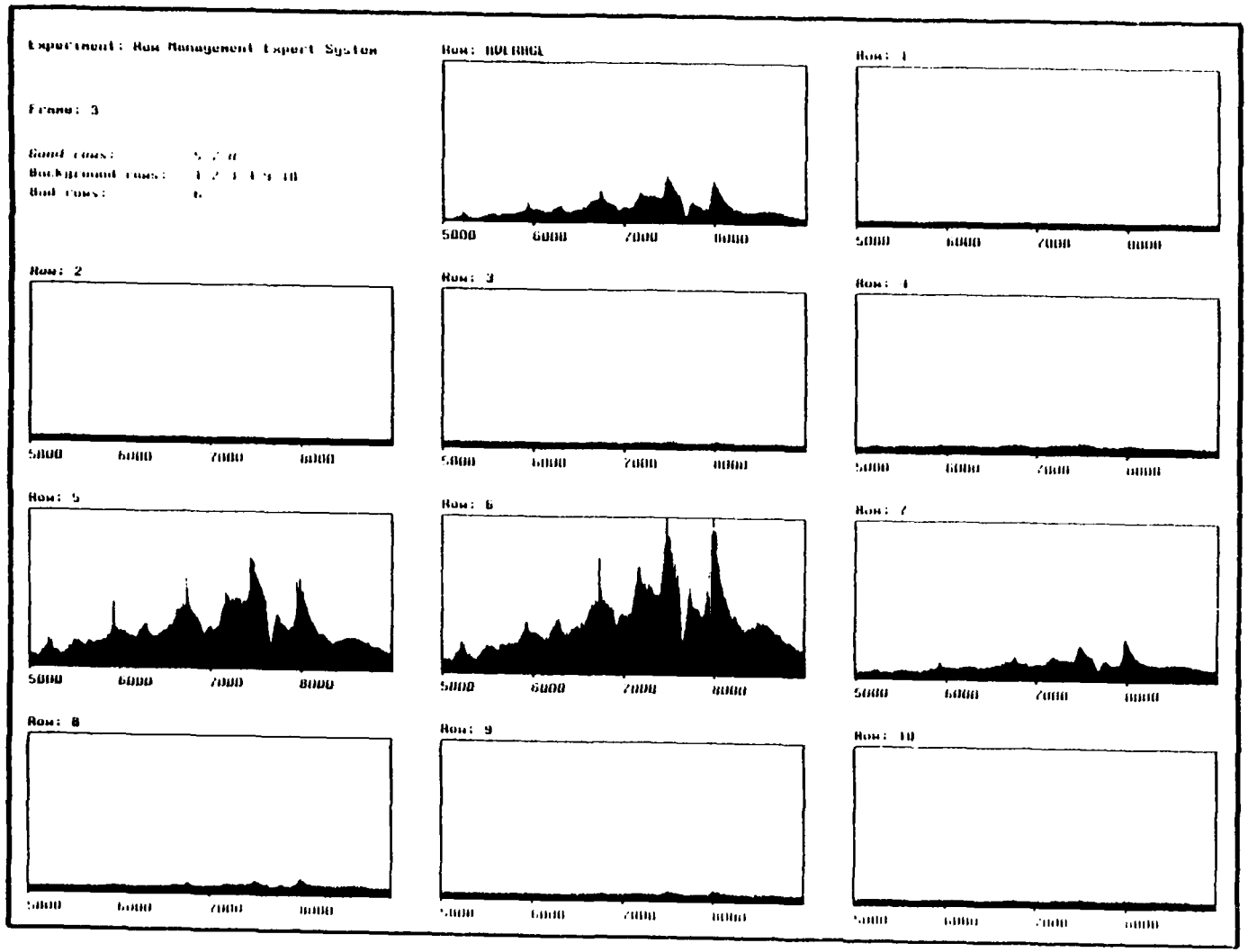

Figure 13. Row Management Frame 3

Frame 4

Facts presented to rule-base:

```
(deffacts average-pixel
     (pixel row  1   9)
     (pixel row  2   9)
     (pixel row  3  11)
     (pixel row  4  21)
     (pixel row  5 118)
     (pixel row  6 126)
     (pixel row  7  49)
     (pixel row  8  21)
     (pixel row  9  16)
     (pixel row 10  14) )

(deffacts area-deviation
     (dev row  1   0)
     (dev row  2   2)
     (dev row  3   4)
     (dev row  4  28)
     (dev row  5 244)
     (dev row  6 244)
     (dev row  7 116)
     (dev row  8  39)
     (dev row  9  26)
     (dev row 10  18) )

(deffacts bad-data
     (fall  row  5)
     (fall  row  6) )
```

Resulting facts from running rule-base:

```
(row 1 background)
(row 2 background)
(row 3 background)
(row 4 data)
(row 7 data)
(row 8 data)
(row 9 data)
(row 10 data)
(row 5 saturated)
(row 6 saturated)
```

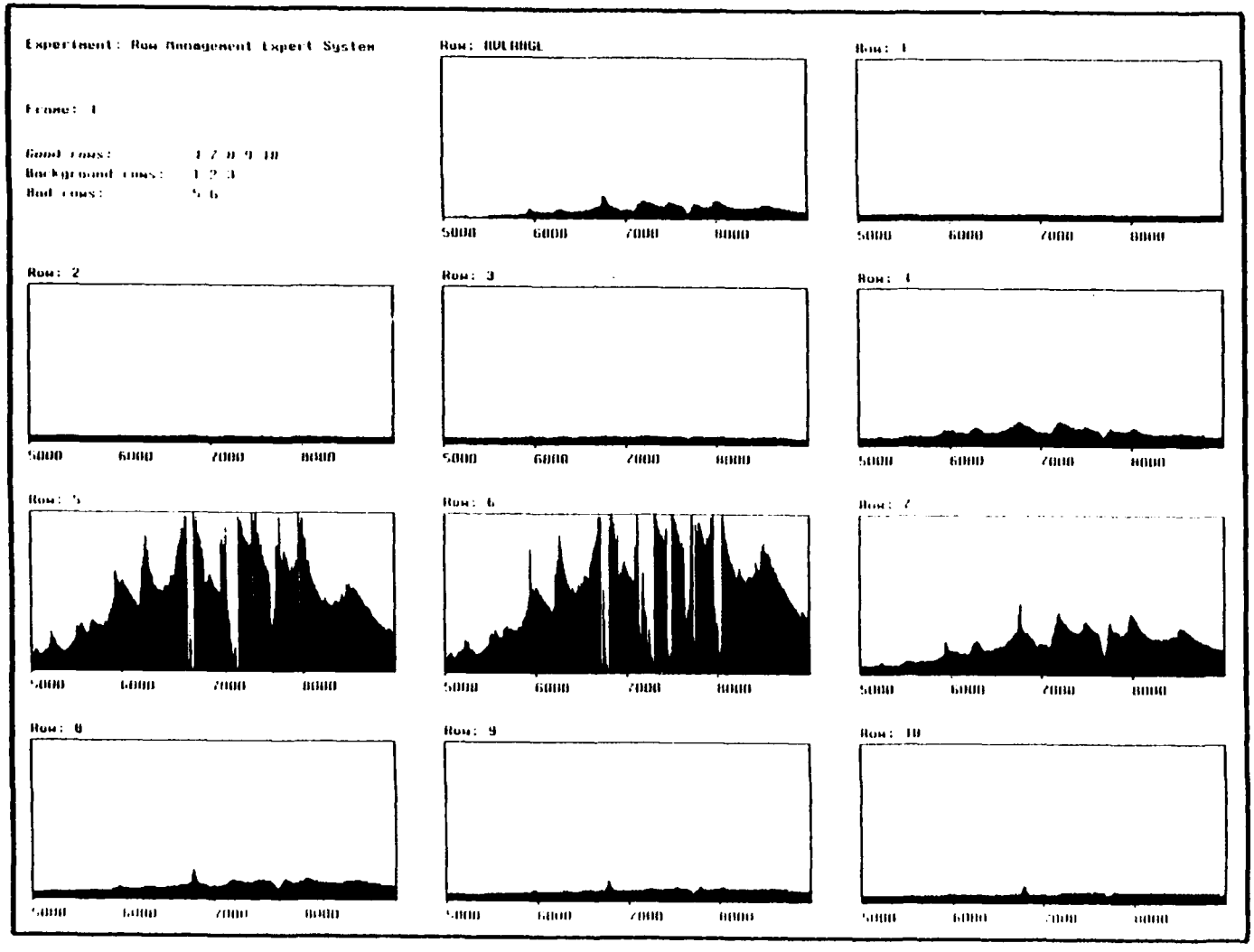

Figure 14. Row Management Frame 4

Frame 5

Facts presented to rule-base:

```
(deffacts average-pixel
     (pixel row  1  11)
     (pixel row  2  11)
     (pixel row  3  14)
     (pixel row  4  33)
     (pixel row  5 153)
     (pixel row  6 216)
     (pixel row  7 195)
     (pixel row  8 141)
     (pixel row  9  94)
     (pixel row 10  55) )

(deffacts area-deviation
     (dev row  1   1)
     (dev row  2   3)
     (dev row  3   8)
     (dev row  4 242)
     (dev row  5 244)
     (dev row  6 240)
     (dev row  7 243)
     (dev row  8 243)
     (dev row  9 243)
     (dev row 10 243) )

(deffacts bad-data
     (fall  row  4)
     (fall  row  5)
     (maxed row  6)
     (fall  row  6)
     (maxed row  7)
     (fall  row  7)
     (maxed row  8)
     (fall  row  8)
     (maxed row  9)
     (fall  row  9)
     (maxed row 10)
     (fall  row 10) )
```

Resulting facts from running rule-base:

(row 1 background)
(row 2 background)
(row 3 background)
(row 4 saturated)
(row 5 saturated)
(row 6 saturated)
(row 7 saturated)
(row 8 saturated)
(row 9 saturated)
(row 10 saturated)

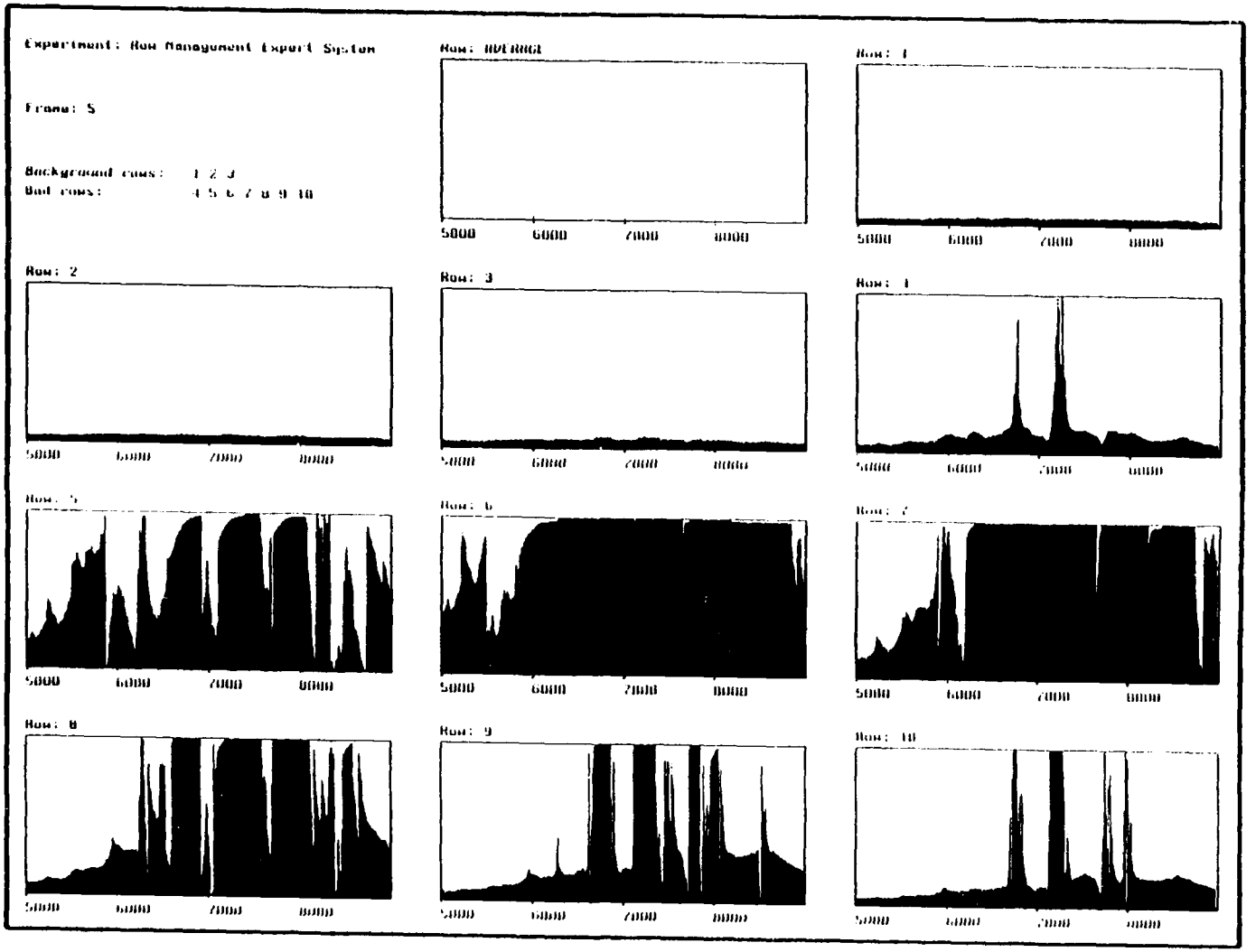

Figure 15. Row Management Frame 5

Frame 6

Facts presented to rule-base:

```
(deffacts average-pixel
     (pixel row  1  10)
     (pixel row  2  10)
     (pixel row  3  11)
     (pixel row  4  13)
     (pixel row  5  39)
     (pixel row  6 107)
     (pixel row  7 123)
     (pixel row  8  97)
     (pixel row  9  32)
     (pixel row 10  18) )

(deffacts area-deviation
     (dev row  1   1)
     (dev row  2   2)
     (dev row  3   4)
     (dev row  4   8)
     (dev row  5  78)
     (dev row  6 241)
     (dev row  7 244)
     (dev row  8 208)
     (dev row  9  42)
     (dev row 10  15) )

(deffacts bad-data
     (fall  row  6)
     (fall  row  7) )
```

Resulting facts from running rule-base:

```
(row 1 background)
(row 2 background)
(row 3 background)
(row 4 background)
(row 5 data)
(row 8 data)
(row 9 data)
(row 10 data)
(row 6 saturated)
(row 7 saturated)
```

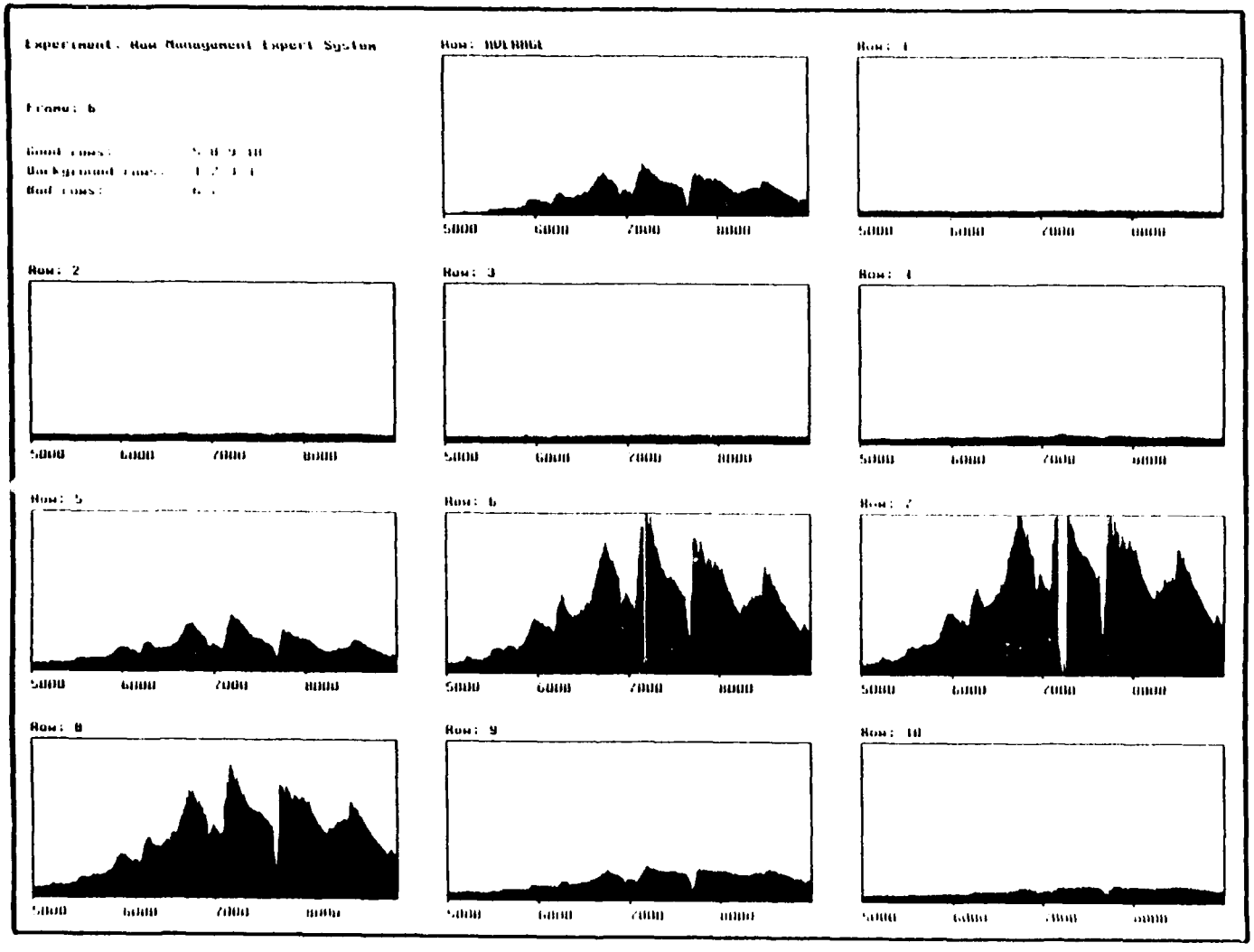

Figure 16. Row Management Frame 6

Frame 7

Facts presented to rule-base:

```
(deffacts average-pixel
     (pixel row  1   9)
     (pixel row  2   9)
     (pixel row  3   9)
     (pixel row  4   9)
     (pixel row  5   9)
     (pixel row  6  11)
     (pixel row  7  15)
     (pixel row  8  11)
     (pixel row  9   9)
     (pixel row 10   9) )

(deffacts area-deviation
     (dev row  1   2)
     (dev row  2   2)
     (dev row  3   1)
     (dev row  4   1)
     (dev row  5   1)
     (dev row  6   4)
     (dev row  7  14)
     (dev row  8   6)
     (dev row  9   1)
     (dev row 10   2) )
```

Resulting facts from running rule-base:

```
(row 1 background)
(row 2 background)
(row 3 background)
(row 4 background)
(row 5 background)
(row 6 background)
(row 7 data)
(row 8 background)
(row 9 background)
(row 10 background)
```

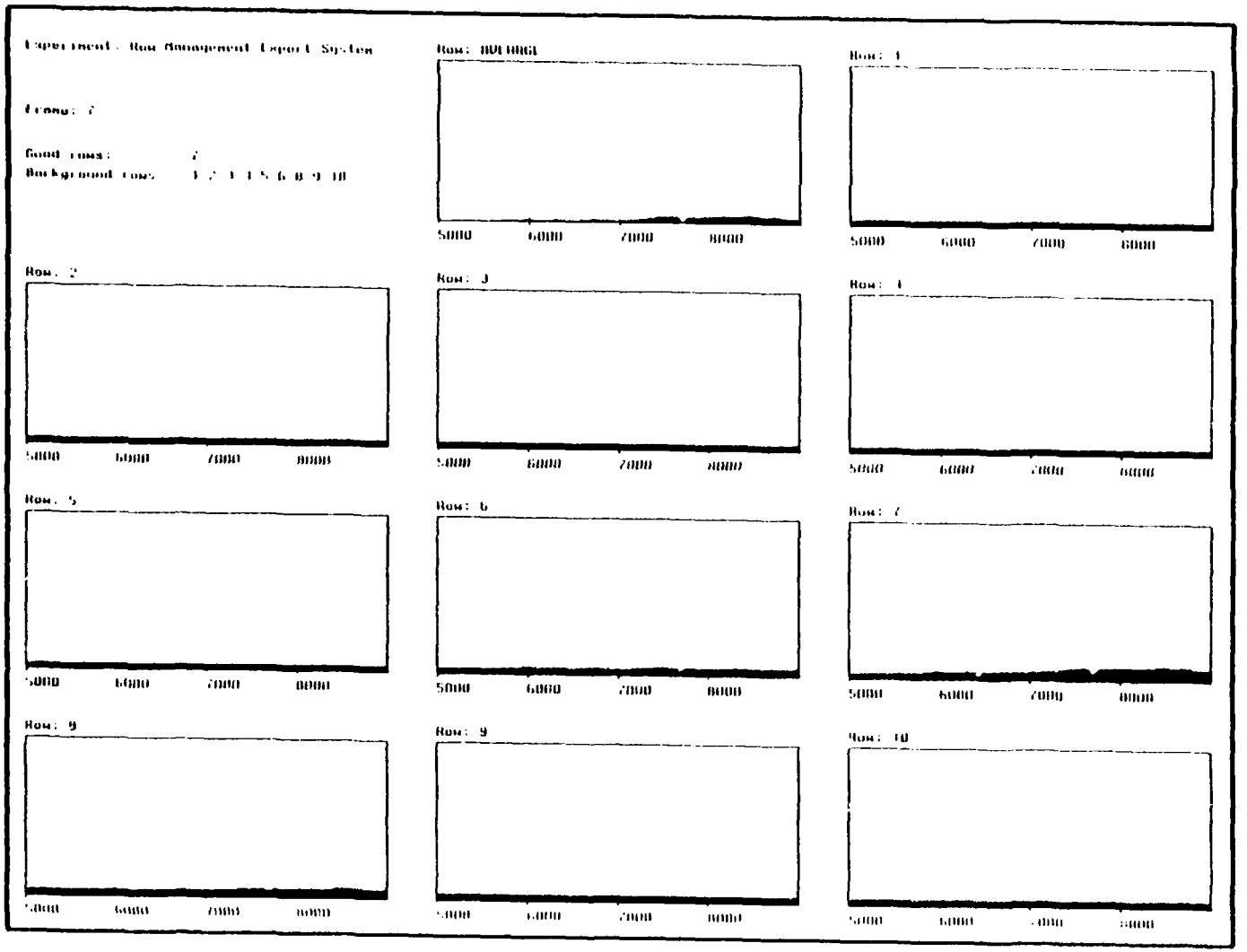

Figure 17. Row Management Frame 7

Frame 8

Facts presented to rule-base:

```
(deffacts average-pixel
     (pixel row  1   9)
     (pixel row  2   9)
     (pixel row  3   9)
     (pixel row  4  10)
     (pixel row  5  12)
     (pixel row  6  40)
     (pixel row  7  96)
     (pixel row  8  50)
     (pixel row  9  14)
     (pixel row 10  10) )

(deffacts area-deviation
     (dev row  1   2)
     (dev row  2   1)
     (dev row  3   1)
     (dev row  4   2)
     (dev row  5   6)
     (dev row  6  70)
     (dev row  7 244)
     (dev row  8  77)
     (dev row  9   9)
     (dev row 10   2) )

(deffacts bad-data
     (fall  row  7) )
```

Resulting facts from running rule-base:

```
(row 1 background)
(row 2 background)
(row 3 background)
(row 4 background)
(row 5 background)
(row 6 data)
(row 8 data)
(row 9 background)
(row 10 background)
(row 7 saturated)
```

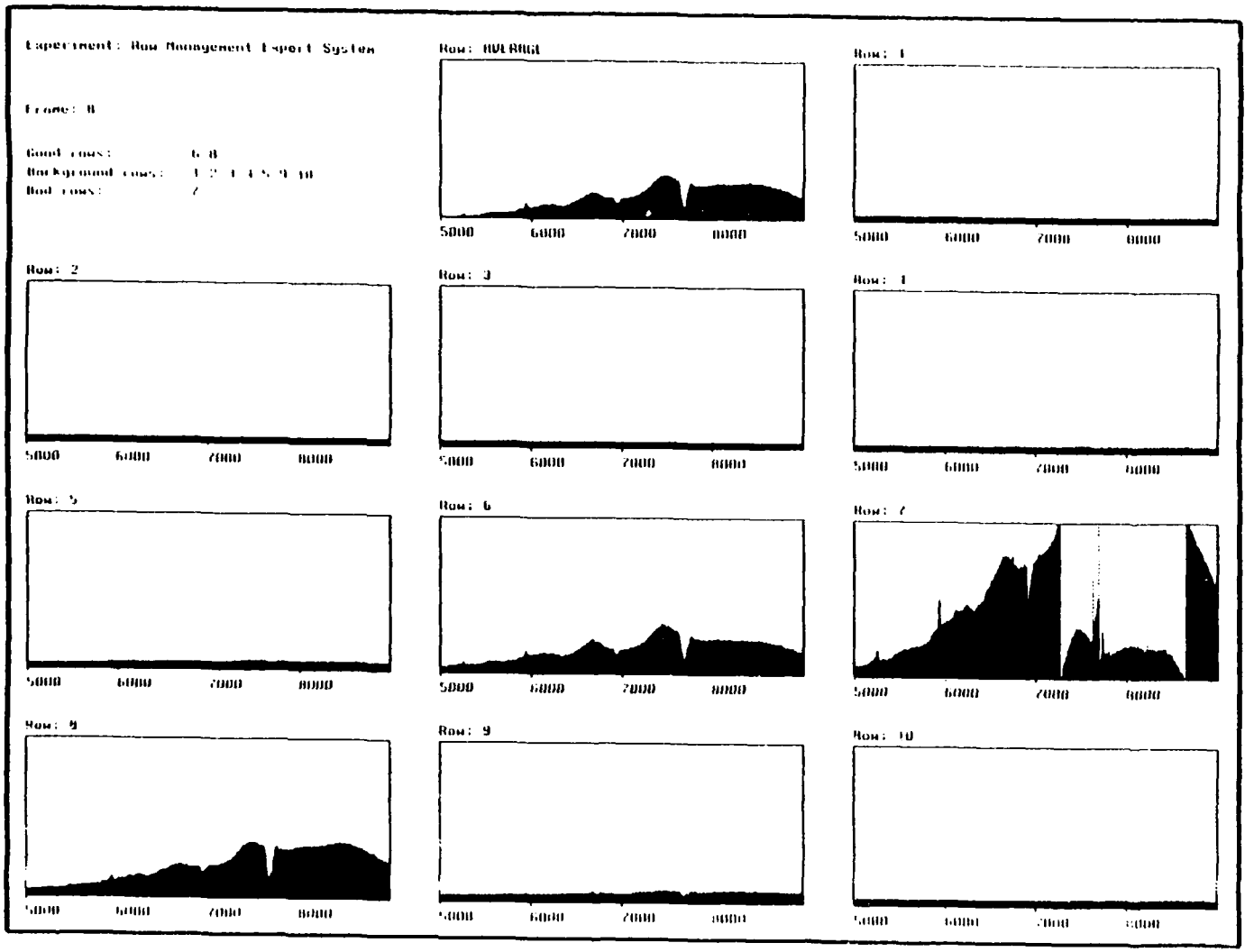

Figure 18. Row Management Frame 8

Frame 9

Facts presented to rule-base:

```
(deffacts average-pixel
     (pixel row  1  26)
     (pixel row  2  54)
     (pixel row  3  63)
     (pixel row  4  74)
     (pixel row  5  83)
     (pixel row  6  34)
     (pixel row  7  22)
     (pixel row  8  60)
     (pixel row  9  66)
     (pixel row 10  50) )

(deffacts area-deviation
     (dev row  1  16)
     (dev row  2  58)
     (dev row  3  76)
     (dev row  4  87)
     (dev row  5 109)
     (dev row  6  31)
     (dev row  7   0)
     (dev row  8  98)
     (dev row  9 108)
     (dev row 10  66) )
```

Resulting facts from running rule-base:

```
(row 1 data)
(row 2 data)
(row 3 data)
(row 4 data)
(row 5 data)
(row 6 data)
(row 7 background)
(row 8 data)
(row 9 data)
(row 10 data)
```

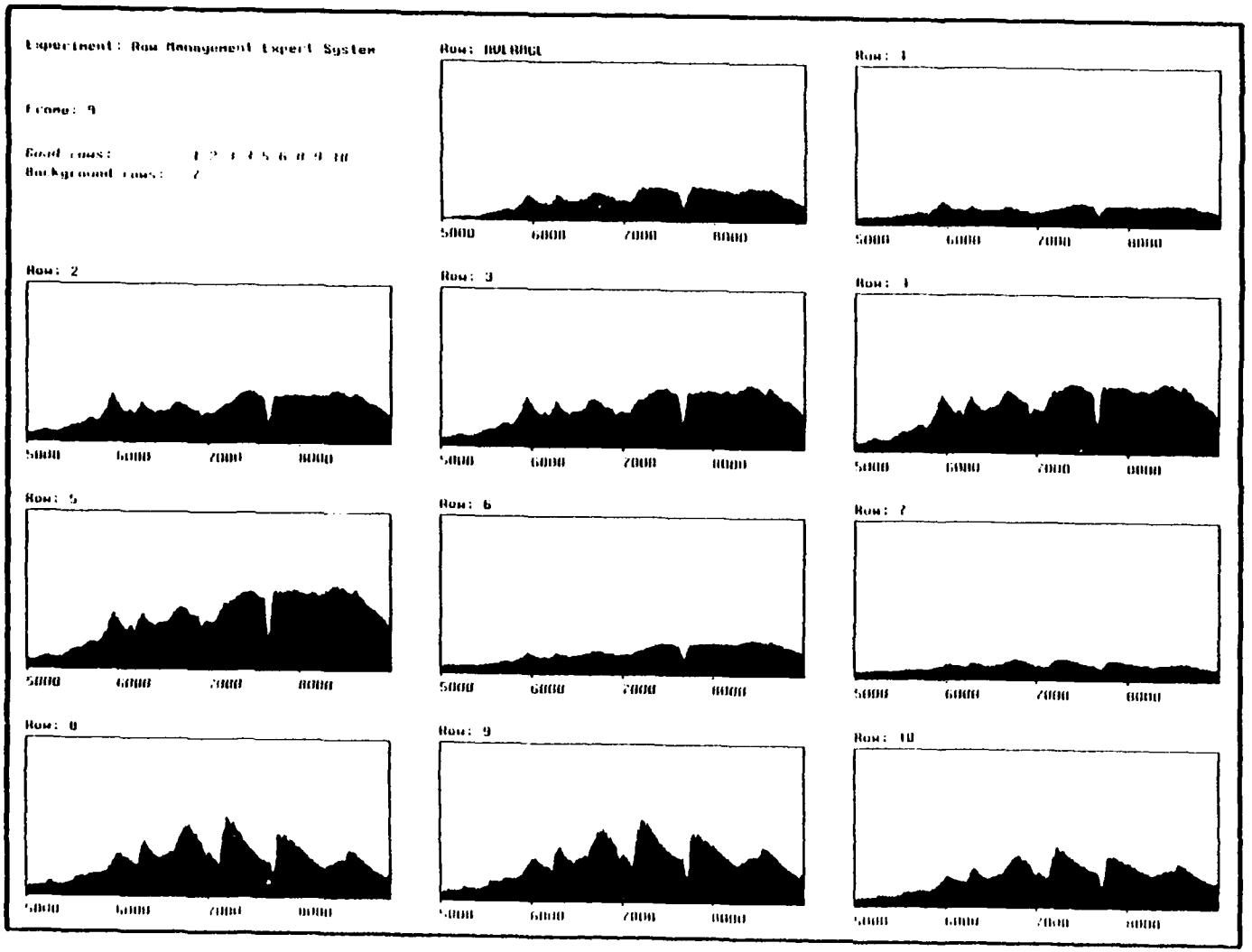

Figure 19. Row Management Frame 9

Frame 10

Facts presented to rule-base:

```
(deffacts average-pixel
     (pixel row  1  26)
     (pixel row  2  26)
     (pixel row  3  26)
     (pixel row  4  26)
     (pixel row  5  26)
     (pixel row  6  26)
     (pixel row  7  26)
     (pixel row  8  26)
     (pixel row  9  26)
     (pixel row 10  26) )

(deffacts area-deviation
     (dev row  1   2)
     (dev row  2   2)
     (dev row  3   1)
     (dev row  4   1)
     (dev row  5   1)
     (dev row  6   1)
     (dev row  7   1)
     (dev row  8   0)
     (dev row  9   1)
     (dev row 10   1) )
```

Resulting facts from running rule-base:

```
(row 1 background)
(row 2 background)
(row 3 background)
(row 4 background)
(row 5 background)
(row 6 background)
(row 7 background)
(row 8 background)
(row 9 background)
(row 10 background)
```

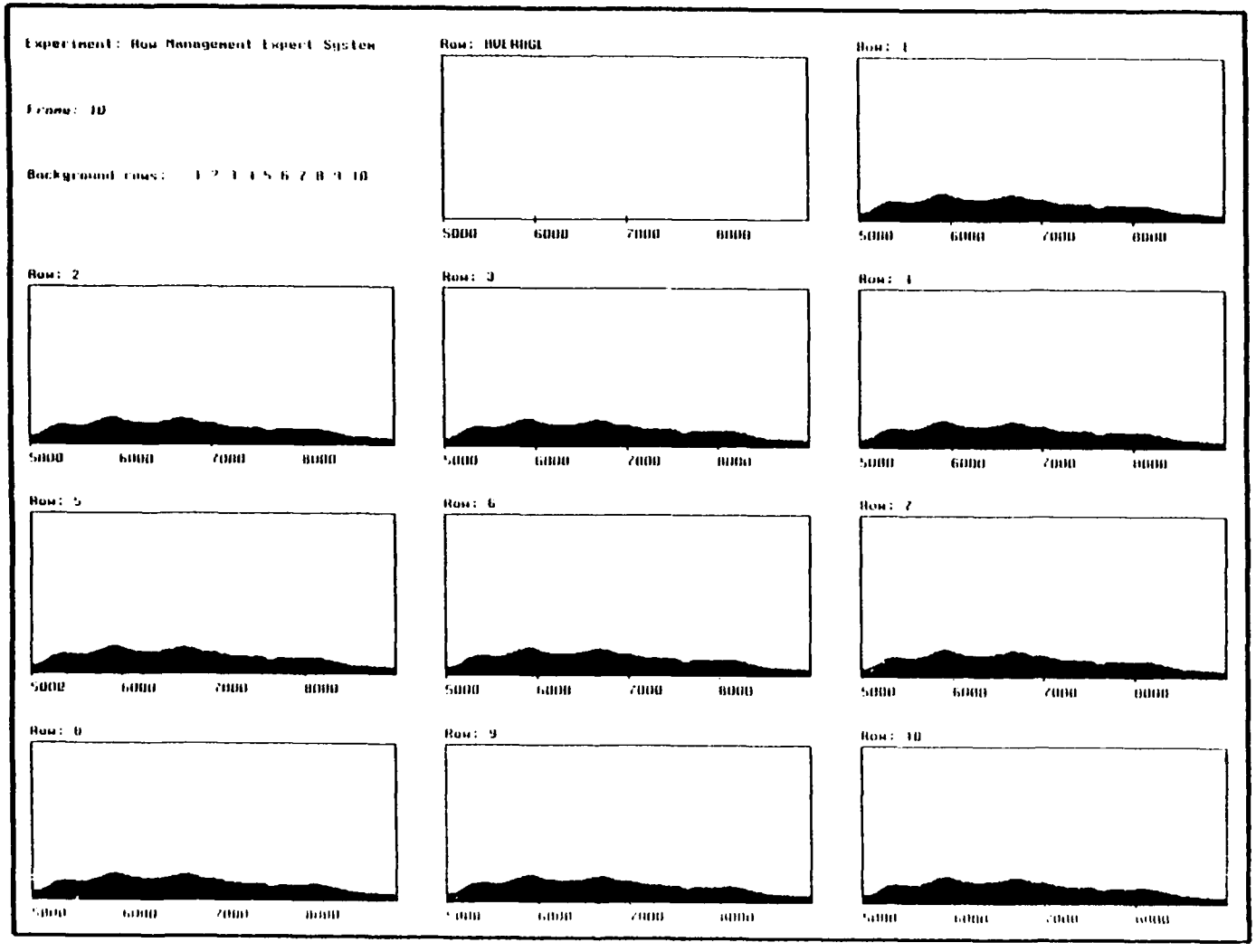

Figure 20. Row Management Frame 10

Frame 11

Facts presented to rule-base:

```
(deffacts average-pixel
     (pixel row  1  27)
     (pixel row  2  27)
     (pixel row  3  27)
     (pixel row  4  29)
     (pixel row  5  30)
     (pixel row  6  27)
     (pixel row  7  27)
     (pixel row  8  26)
     (pixel row  9  27)
     (pixel row 10  27) )

(deffacts area-deviation
     (dev row  1   1)
     (dev row  2   2)
     (dev row  3   9)
     (dev row  4  38)
     (dev row  5  18)
     (dev row  6   5)
     (dev row  7   1)
     (dev row  8   0)
     (dev row  9   1)
     (dev row 10   1) )
```

Resulting facts from running rule-base:

```
(row 1 background)
(row 2 background)
(row 3 background)
(row 4 data)
(row 5 data)
(row 6 background)
(row 7 background)
(row 8 background)
(row 9 background)
(row 10 background)
```

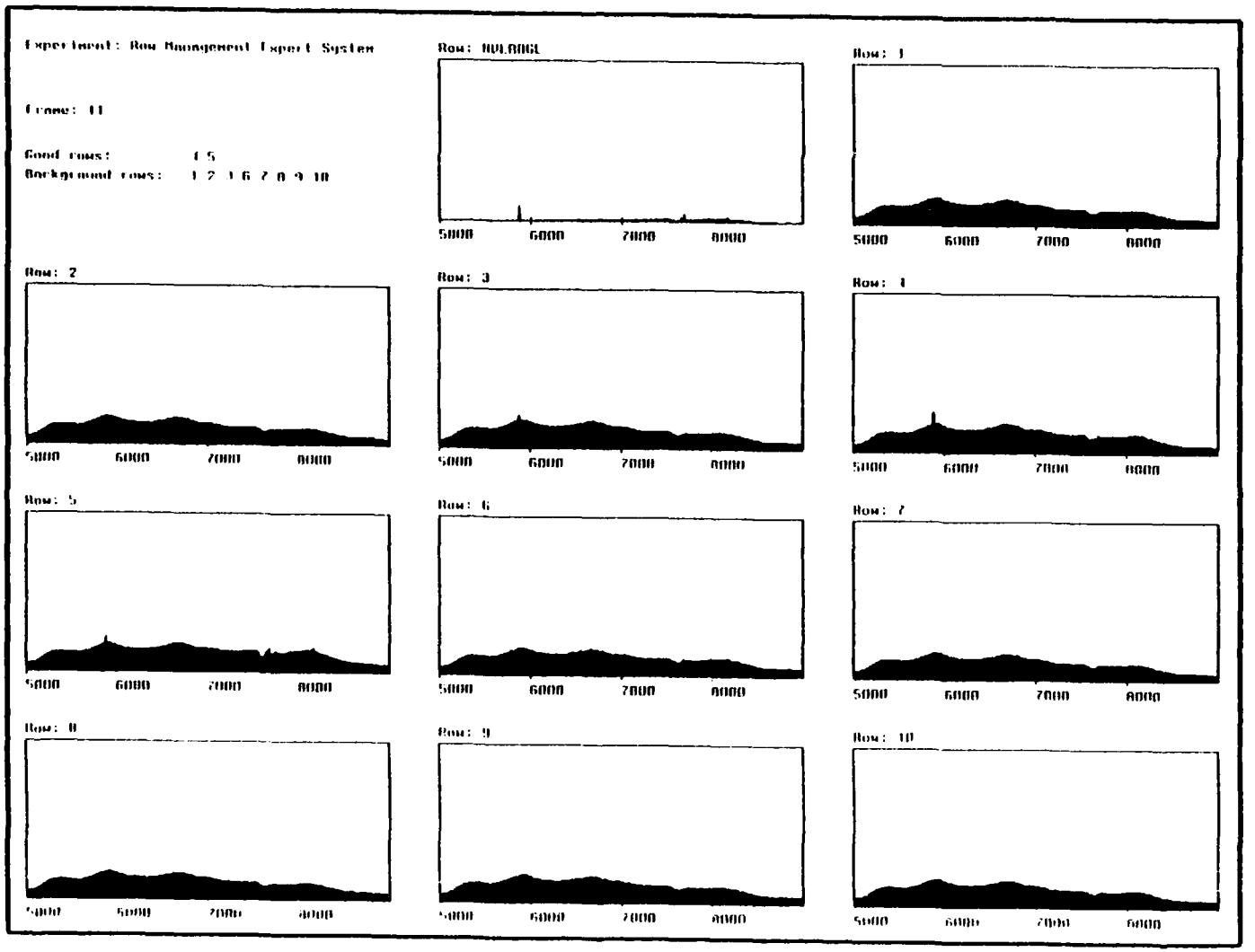

Figure 21. Row Management Frame 11

Frame 12

Facts presented to rule-base:

```
(deffacts average-pixel
     (pixel row  1  37)
     (pixel row  2  37)
     (pixel row  3  39)
     (pixel row  4  42)
     (pixel row  5  45)
     (pixel row  6  40)
     (pixel row  7  37)
     (pixel row  8  36)
     (pixel row  9  37)
     (pixel row 10  37) )

(deffacts area-deviation
     (dev row  1   2)
     (dev row  2   4)
     (dev row  3  19)
     (dev row  4  53)
     (dev row  5  46)
     (dev row  6  12)
     (dev row  7   3)
     (dev row  8   0)
     (dev row  9   2)
     (dev row 10   2) )
```

Resulting facts from running rule-base:

```
(row 1 background)
(row 2 background)
(row 3 data)
(row 4 data)
(row 5 data)
(row 6 data)
(row 7 background)
(row 8 background)
(row 9 background)
(row 10 background)
```

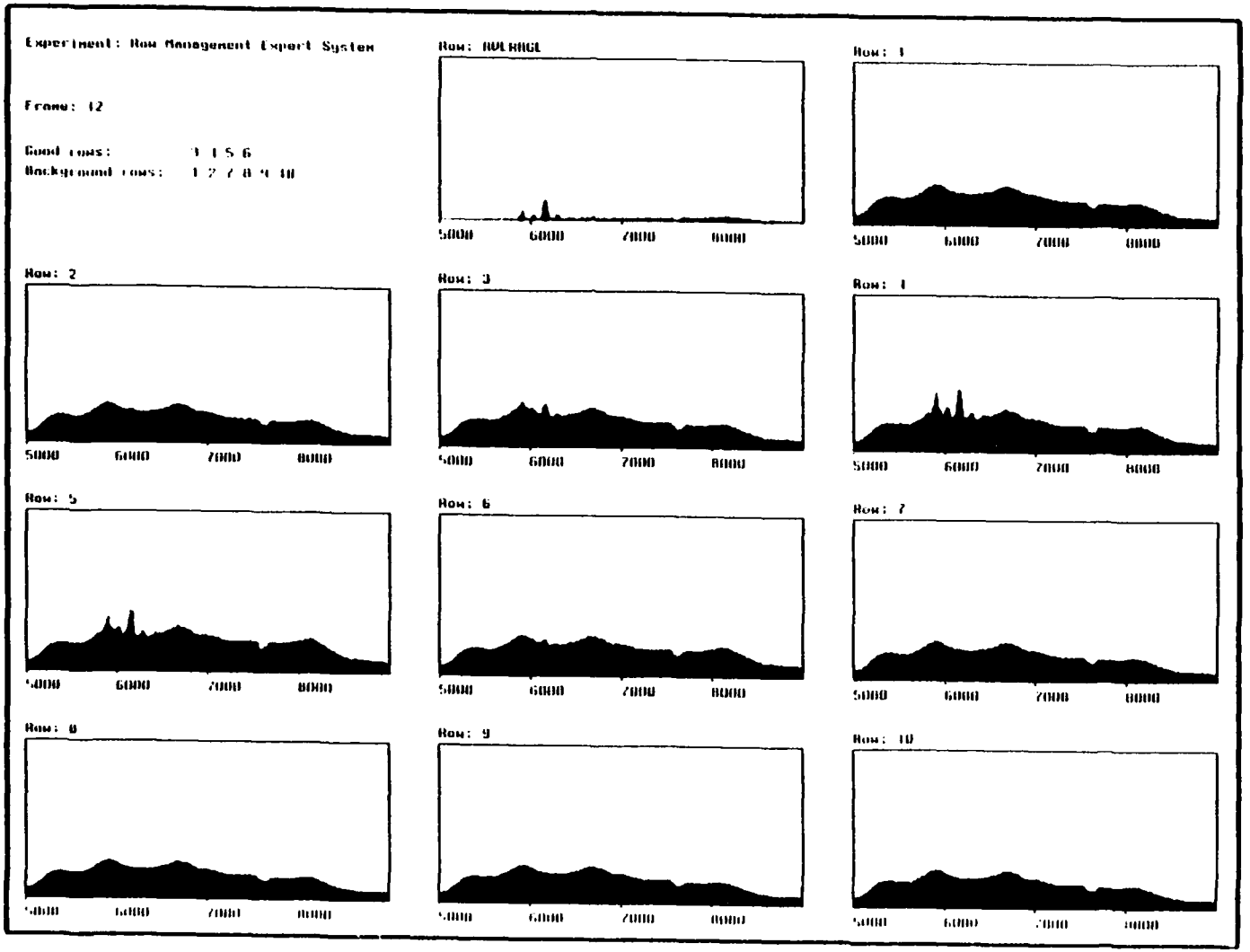

Figure 22. Row Management Frame 12

Frame 13

Facts presented to rule-base:

```
(deffacts average-pixel
     (pixel row  1  64)
     (pixel row  2  69)
     (pixel row  3  83)
     (pixel row  4  88)
     (pixel row  5  68)
     (pixel row  6  61)
     (pixel row  7  60)
     (pixel row  8  60)
     (pixel row  9  60)
     (pixel row 10  60) )

(deffacts area-deviation
     (dev row  1  26)
     (dev row  2  72)
     (dev row  3 159)
     (dev row  4 139)
     (dev row  5  56)
     (dev row  6   5)
     (dev row  7   2)
     (dev row  8   0)
     (dev row  9   2)
     (dev row 10   2) )
```

Resulting facts from running rule-base:

```
(row 1 data)
(row 2 data)
(row 3 data)
(row 4 data)
(row 5 data)
(row 6 background)
(row 7 background)
(row 8 background)
(row 9 background)
(row 10 background)
```

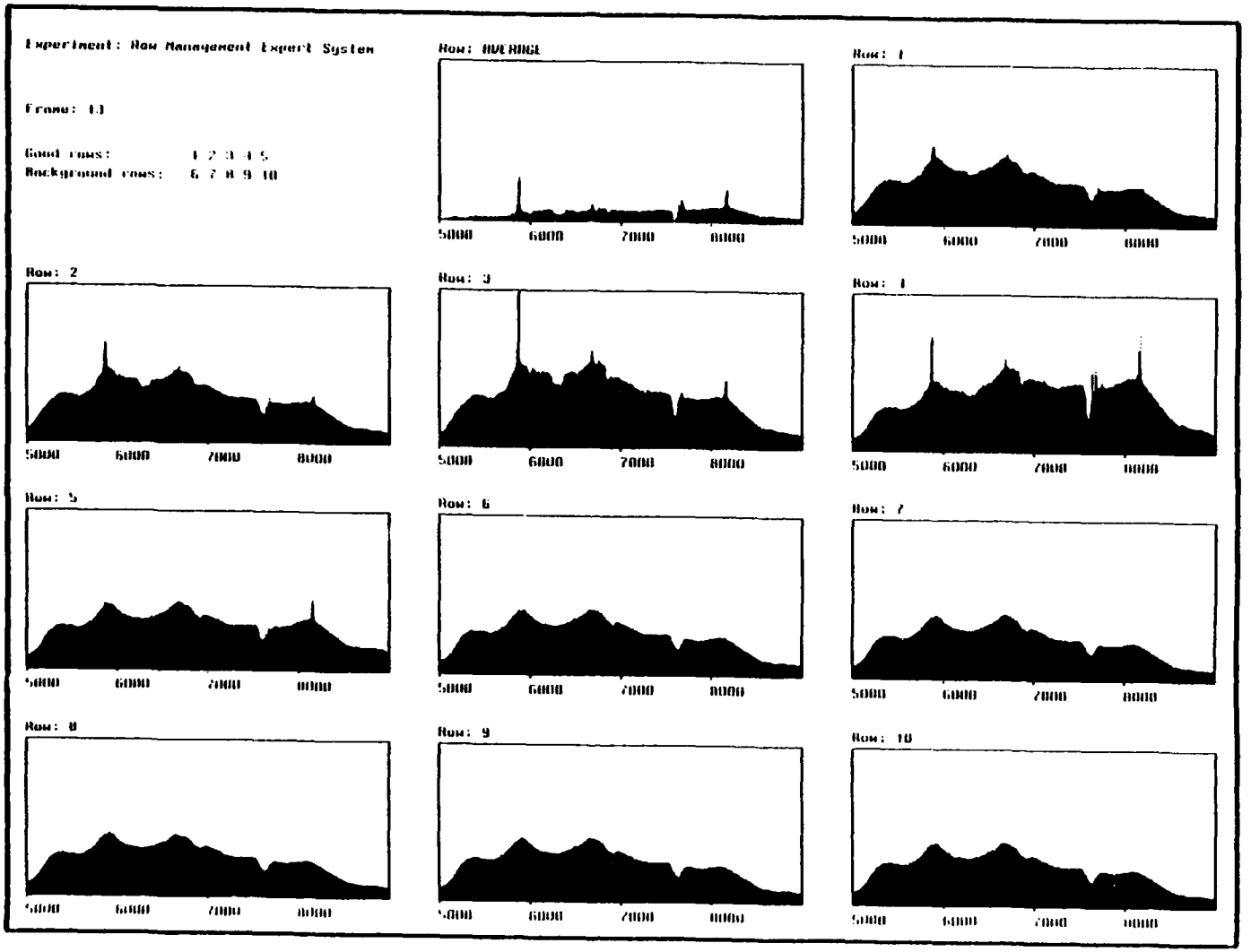

Figure 23. Row Management Frame 13

Frame 14

Facts presented to rule-base:

```
(deffacts average-pixel
     (pixel row  1  82)
     (pixel row  2  98)
     (pixel row  3 141)
     (pixel row  4 144)
     (pixel row  5  91)
     (pixel row  6  79)
     (pixel row  7  77)
     (pixel row  8  77)
     (pixel row  9  77)
     (pixel row 10  77) )

(deffacts area-deviation
     (dev row  1  34)
     (dev row  2 124)
     (dev row  3 140)
     (dev row  4 188)
     (dev row  5  82)
     (dev row  6   7)
     (dev row  7   3)
     (dev row  8   0)
     (dev row  9   2)
     (dev row 10   2) )

(deffacts bad-data
     (maxed row  3)
     (maxed row  4) )
```

Resulting facts from running rule-base:

```
(row 1 data)
(row 2 data)
(row 5 data)
(row 6 background)
(row 7 background)
(row 8 background)
(row 9 background)
(row 10 background)
(row 3 saturated)
(row 4 saturated)
```

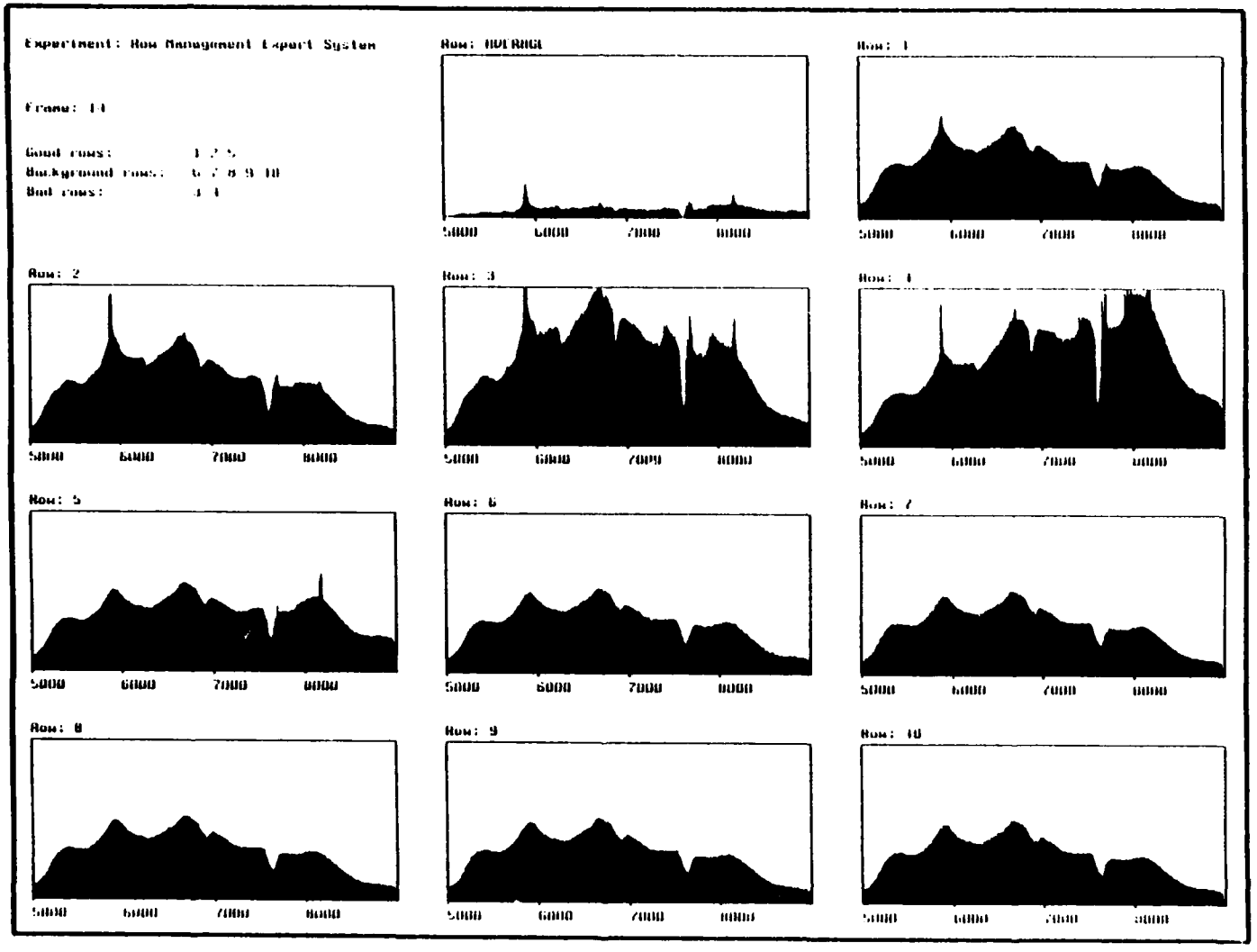

Figure 24. Row Management Frame 14

Frame 15

Facts presented to rule-base:

```
(deffacts average-pixel
     (pixel row  1 117)
     (pixel row  2 117)
     (pixel row  3 117)
     (pixel row  4 117)
     (pixel row  5 117)
     (pixel row  6 116)
     (pixel row  7 116)
     (pixel row  8 115)
     (pixel row  9 115)
     (pixel row 10 116) )

(deffacts area-deviation
     (dev row  1   6)
     (dev row  2  12)
     (dev row  3   5)
     (dev row  4   6)
     (dev row  5   5)
     (dev row  6   6)
     (dev row  7   3)
     (dev row  8   1)
     (dev row  9   3)
     (dev row 10   3) )
```

Resulting facts from running rule-base:

```
(row 1 background)
(row 2 background)
(row 3 background)
(row 4 background)
(row 5 background)
(row 6 background)
(row 7 background)
(row 8 background)
(row 9 background)
(row 10 background)
```

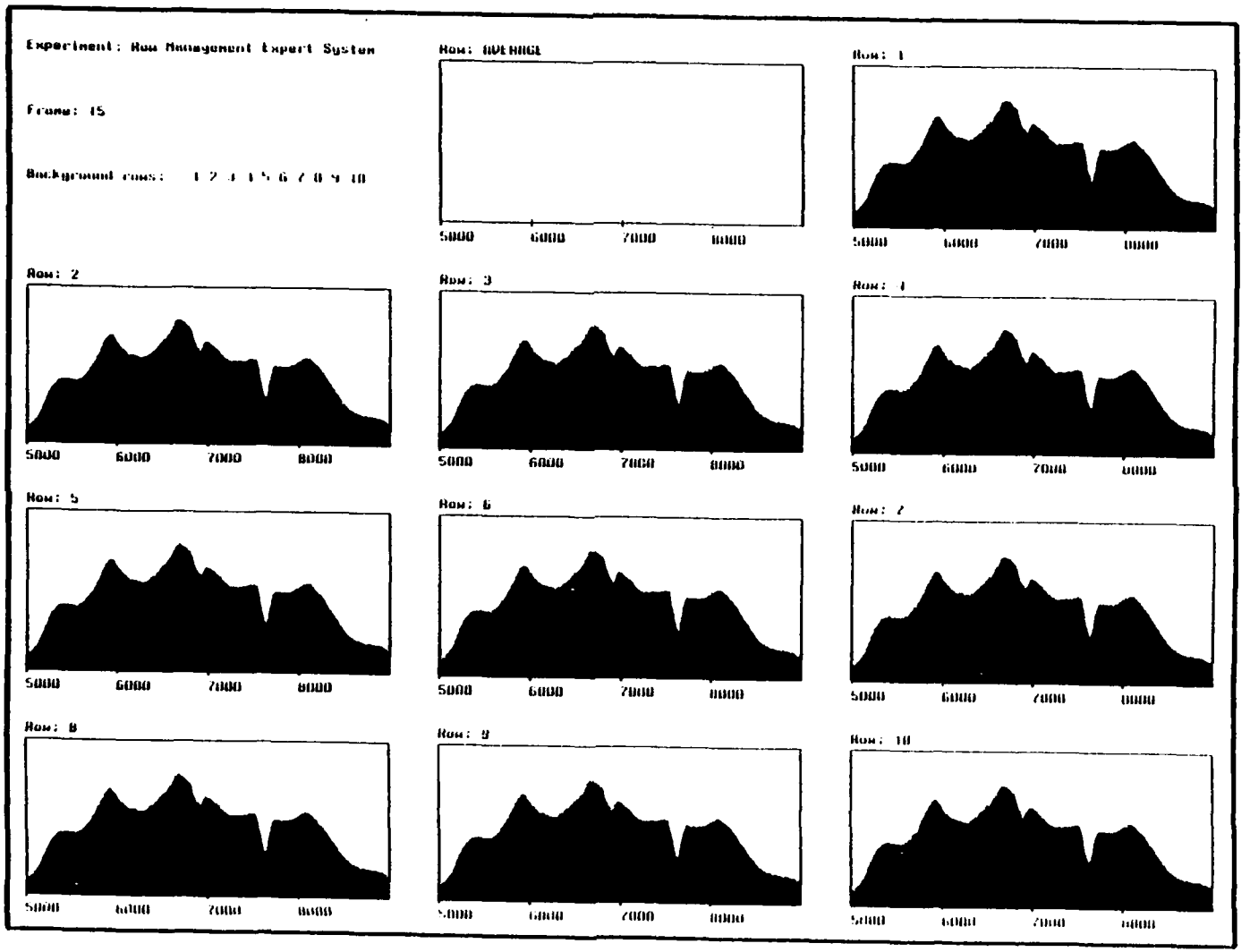

Figure 25. Row Management Frame 15

Bibliography

1. Barnard, Etienne and David P. Casasent. "Optical Neural Net for Classifying Imaging Spectrometer Data," Applied Optics, 28: 3129-3133 (1 August 1989).

2. Botha, Elizabeth and others. "Optical Neural Networks for Image Analysis: Imaging Spectroscopy and Production Systems," IEEE International Conference on Neural Networks, I: 541-546. San Diego: SOS Printing, 1988.

3. Caulfield, H. John and others. "Optical Neural Networks," Proceedings of the IEEE, 77: 1573-1583 (October 1989).

4. Draper, James S. and others. "A Microcomputer Neural Net Benchmark Against Standard Classification Techniques," IEEE First International Conference on Neural Networks, IV: 651-658. San Diego: SOS Printing, 1987.

5. Harper, Alice M. and Shirley A. Liebman. "Intelligent Instrumentation," Journal of Research of the National Bureau of Standards, 90: 453-464 (November-December 1985).

6. Johnson Space Center. CLIPS Reference Manual. JSC-22948. Houston: NASA, July 1989.

7. Kabrisky, Matthew. Class lectures in EENG 620, Pattern Recognition I. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, January 1989.

8. Kabrisky, Matthew and Steven K. Rogers. "Lectures on Pattern Recognition." Class handout distributed in EENG 620, Pattern Recognition I. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, January 1989.

9. Lippmann, Richard P. "An Introduction to Computing with Neural Nets," IEEE Acoustics, Speech, and Signal Processing (ASSP) Magazine, 4-22 (April 1987).

10. Micro-Search for IBM-PC and Compatibles. Product brochure. Sprouse Scientific Systems, Inc., Paoli PA, undated.

11. Miller, Dr L. Grant, Chief, Electro-Optics Branch. Personal interviews. Wright-Patterson AFB OH, 17 January through 19 May 1989.

12. Petty, Charles C. "Spectroscopy Applications." Class handout distributed in Spectroscopy Applications. Wright-Patterson AFB OH, September 1980.

13. Rogers, Maj Steven K., Associate Professor, School of Engineering. Personal interviews. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 13 January 1989 through 2 April 1990.

14. Rogers, Steven K. and Matthew Kabrisky. "Biological and Artificial Neural Networks for Pattern Recognition." Class handout distributed in EENG 620, Pattern Recognition I. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1989.

15. Ruck, Dennis W. and others. "Feature Selection Using a Multilayer Perceptron." School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 7 November 1989. Accepted for publication in The Journal of Neural Network Computing, Fall 1990.

16. Ruck, Dennis W. and others. "The Multilayer Perceptron: A Bayes Optimal Discriminant Function Approximator." School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 15 March 1990. Submitted for publication in IEEE Transactions on Neural Networks.

17. Sadlier, Charles J. and John W. Leahey. Program for Digital Computer Determination of Spectral Lines. Contract AF33(600)-42825. Lowell MA: Lowell Technological Institute Research Foundation, 6 December 1962.

18. Sadtler PC Spectral Search Libraries. Product Brochure. Sadtler Research Laboratories, Philadelphia PA, undated.

19. Stone, M. "Cross-Validatory Choice and Assessment of Statistical Prediction," Journal of the Royal Statistical Society Series B, 36: 111-133.

20. Wilkins, Charles L. "Interactive Pattern Recognition in the Chemical Analysis Laboratory," Journal of Chemical Information and Computer Sciences, 17: 242-249 (1977).

Vita

First Lieutenant Arthur L. [redacted]. He graduated from high school in Tifton, Georgia, in 1983 and attended the United States Air Force Academy, from which he received the degree of Bachelor of Science in Computer Science in May 1987. Upon graduation, he received a commission in the USAF. He was temporarily assigned to the Texas Prefreshman Engineering Program (TexPrep) at the Pan American University in Edinburg, Texas as a computer science instructor during the summer before reporting to his first assignment. In August of 1987, he reported to Wright-Patterson AFB as an Electro-Optics Engineer. In January 1988, he began his masters degree at AFIT as a part-time student.

[redacted]

2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	Approved for public release; distribution unlimited
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
AFIT/GCE/ENG/90J-01	

6a NAME OF PERFORMING ORGANIZATION	6b OFFICE SYMBOL *(If applicable)*	7a NAME OF MONITORING ORGANIZATION
School of Engineering	AFIT/ENG	
6c ADDRESS (*City, State, and ZIP Code*)		7b ADDRESS (*City, State, and ZIP Code*)
Air Force Institute of Technology (AU) Wright-Patterson AFB, Ohio 45433-6583		
8a NAME OF FUNDING / SPONSORING ORGANIZATION	8b OFFICE SYMBOL *(If applicable)*	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (*City, State, and ZIP Code*)		10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (*Include Security Classification*)
INTEGRATING RULE-BASED AND NEURAL-NET TECHNIQUES FOR SPECTRAL ANALYSIS

12 PERSONAL AUTHOR(S)
Arthur L. Sumner, 1Lt, USAF

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (*Year, Month, Day*)	15 PAGE COUNT
MS Thesis	FROM _____ TO _____	1990 June	108

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (*Continue on reverse if necessary and identify by block number*)
FIELD	GROUP	SUB-GROUP	
12	09		Visible Spectra, Pattern Recognition, Neural Nets, Spectrum Analysis, Atomic Spectroscopy, Molecular Spectroscopy, Artificial Intelligence
20	05		

19 ABSTRACT (*Continue on reverse if necessary and identify by block number*)

Thesis Advisor: Steven K. Rogers
Associate Professor
Department of Electrical and Computer Engineering

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (*Include Area Code*)	22c OFFICE SYMBOL
Steven K. Rogers, Assoc. Professor	(513) 225-9266	ENG

DD Form 1473, JUN 86 *Previous editions are obsolete*

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Spectral analysis involving the determination of atomic and molecular species present in multi-spectral data is a very time consuming task, especially considering the fact that there are typically thousands of spectra collected during each experiment. Due to the overwhelming amount of available spectral data and the time required to analyze these data, a robust automatic method for performing preliminary spectral analysis is needed. This research focused on the development of a rule-based expert system and a supervised artificial neural network with error correction learning, specifically a three-layer, feed-forward, back-propagation perceptron. The objective was to develop an integrated spectral analysis system which would perform preliminary spectral analysis and save the analysts from the task of reviewing thousands of spectral frames. The input to the neural network, which is screened by the rule-base, is raw spectral data, with the output consisting of the classification of both atomic and molecular species in the source.

The idea that computers can perform the spectral identification of molecular and atomic radiators goes back to the earliest days of computers. Most commercially available spectral identification programs use some type of statistical pattern recognition technique containing a database of known species. One of the greatest advantages of neural networks over traditional pattern recognition techniques is the ability of the neural networks to learn. Once a neural network has been trained, it may be able to make classifications with a general amount of robustness not normally found in statistical recognizers. Another major advantage of neural networks over statistical methods is that new species can be recognized by simply presenting samples of spectra to the neural network, eliminating the time-consuming task of trying to statistically characterize the new species. Neural networks are not able to recognize every species and will never replace the analyst, but they can reduce the time required for preliminary analysis and aid the analyst in picking out species that a statistical technique might have missed.